ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(НИУ «БелГУ»)

Институт межкультурной коммуникации и международных отношений

Кафедра английской филологии и межкультурной коммуникации

Создание веб-ресурса лаборатории языкового мониторинга

Выпускная квалификационная работа
обучающегося по направлению подготовки
45.03.04 Интеллектуальные системы в гуманитарной сфере
очной формы обучения,
группы 04001320
Никифорова Антона Валерьевича

Научный руководитель
профессор, доктор
филологических наук
Куприева И.А.

БЕЛГОРОД 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Абсолютно новые слои населения начинают пользоваться компьютером с появлением и развитием Web-технологий. Глобальная Сеть уже настолько основательно вошла в нашу жизнь, что публикация данных в WWW стала обыденностью. Вероятно, какое-то время назад, когда для существенной части общества Интернет был чем-то новым и не совсем ясным и доступным, вопрос «для чего все это необходимо» казался безусловно логичным.

Web-технология на все сто процентов поменяла наши взгляды и понятия о работе с информацией, да и с компьютером в целом. По всей видимости основные характеристики развития вычислительной техники – производительность, быстродействие, емкость памяти – не принимали во внимание основного «узкого места» системы – интерфейса с человеком. Устаревшая система взаимодействия человека с информационной системой ограничивал внедрение новых технологий и снижал выгоду от их использования. И только лишь когда интерфейс между человеком и компьютером был упрощен до естественности восприятия простым человеком, произошел небывалый взрыв интереса к потенциалу вычислительной техники.

Люди начали осознавать, что компьютер – это не просто модная и дорогостоящая игрушка, но инструмент извлечения важной и актуальной информации. При этом им не надо было тратить кучу времени на освоение технологии работы с компьютером (в сравнении с тем, как это было раньше).

За счет пользователей, не имеющих отношения к категории экспертов в сфере информационных технологий, все время расширяется диапазон социальных групп, подключающихся и пользующихся бескрайними

возможностями сети Интернет. Это доктора, спасатели, спортсмены, учителя, писатели, актеры, духовные лица, строители, менеджеры, художники, адвокаты. Перечень можно продолжать бесконечно долго. Каждый, кто почувствовал выгодность и необходимость использования Сети для своей деятельности или увлечения, примыкает к безграничной армии потребителей данных во «Всемирной Паутине».

С развитием технологий гипертекстовой разметки в сети Интернет начало возникать все больше сайтов, тематика которых была абсолютно разной – от сайтов крупных фирм, рассказывающих о преуспевании фирмы, до сайтов небольших компаний, предлагающих посетить их офисы. Развитие стало толчком к появлению сайтов, даже целых порталов, на которых абсолютно все могут общаться и получать ответы на всевозможные вопросы.

В настоящее время практически каждая организация имеет свой web-сайт. Это необходимый фактор существования, позволяющий привлечь новых клиентов, рассказать о своей работе, поделиться опытом и развить свое дело. Все эти факторы крайне важны для Научно-исследовательской лаборатории языкового мониторинга (НИЛЯМ), так как она изучает вариативность английского языка Великобритании в условиях влияния различных политических, социальных и экономических факторов. Одно из первенствующих направлений исследования – это вопросы влияния языков иммигрантов на английский язык.

Согласно словам исследователей, актуальная ситуация в Великобритании, куда каждый год приезжает большое число мигрантов из разных стран, дает возможность собрать достаточный объем данных. Адаптируясь к новым условиям, иммигранты не всегда рвутся ассимилироваться полностью: они сохраняют собственную культуру, тем самым оказывая неминуемое воздействие на принимающую культуру, в частности, на язык. Новая лаборатория изучает языковые ситуации Великобритании в условиях мультикультурализма.

В 2017 году эксперты будут анализировать публикации в онлайн-СМИ и посты в социальных сетях. Они собираются систематизировать примеры языковых изменений в данных материалах и сформировать базу данных семантических сдвигов лексики на нынешней стадии развития английского языка в Великобритании.

Вдобавок ученые работают над программой углубленного изучения текстов виртуальной реальности. Она даст возможность обрабатывать большие текстовые массивы и отмечать языковые изменения с учетом развития нынешней социополитической ситуации в мире.

Таким образом, актуальность данной работы заключается в необходимости создания веб-ресурса, который позволит пользователям Интернета получить необходимую информацию по работе лаборатории.

Объектом исследования данной работы является анализ информационных технологий при разработке сайта.

Предметом исследования является разработка web-сайта.

Данная работа включает различные методы исследования, такие как анализ литературы, сравнение технологий разработки, изучение и обобщение практики разработчиков сайтов и проведение теоретического анализа.

Основной целью данного исследования является освещение технологий разработки и непосредственно создание web-сайта для Научно-исследовательской лаборатории языкового мониторинга.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить теоретический материал по темам «Этапы создания веб-ресурса», «Проектирование баз данных», «Разработка web-сайта»;
- провести анализ технологий разработки проекта;
- спроектировать базу данных;

- выполнить программирование моделей, шаблонов и контроллеров.

Решение вышеперечисленных задач позволит реализовать основную цель проекта и приведет к созданию полностью готового сайта НИЛЯМ.

Структура данной работы включает введение, две главы, заключение и список используемых источников.

Глава 1 Анализ предметной области и технологий разработки

1.1 Понятие web-сайта

Существенная часть информации, доступная пользователям Интернета, организована в виде web-сайтов. Каждый из них имеет собственное имя (адрес) в Интернете.

Web-сайт – это информация, показанная в определенном виде, которая располагается на web-сервере. Для просмотра web-сайтов на компьютере пользователя используются специальные программы, называемые браузерами.

Любая страница web-сайта также имеет собственный Интернет адрес, который состоит из адреса сайта и имени файла, принадлежащего этой странице. Таким образом, web-сайт – это информационный источник, состоящий из связанных между собой гипертекстовых документов (web-страницы), размещенный на web-сервере и имеющий персональный адрес. Посмотреть web-сайт может каждый человек, имеющий компьютер, подключенный к Интернету.

1.2 Этапы разработки web-сайта

Первым этапом разработки веб-ресурса является составление и утверждение технического задания на разработку сайта. После этого следует проектирование базы данных (БД). Следом идет написание моделей базы данных (методы ввода/вывода данных таблиц БД) и написание шаблонов. После происходит написание методов контроллеров, которые связывают данные и представление. В конце работы создается документации исходного кода проекта и происходит тестирование и отладка функционала (Колисниченко, 2013).

1.3 Основные функции НИЛЯМ

Ниже перечислены главные функции Научно-исследовательской лаборатории языкового мониторинга.

- Исследование факторов, влияющих на эволюцию лингвокультуры, изучение характера их влияния.
- Сбор и лингвистическая обработка релевантного фактического материала, создание баз данных.
- Разработка программы углубленного эмпирического исследования, в которой будут отражены релевантные методы для проведения мониторинга языковой ситуации в избранном аспекте, а также для

изучения языковых изменений с учетом развития текущей социополитической ситуации в мире. Создание уровневой модели языковых инвариантов и вариантов.

- Разработка мультимедийный лингводидактический тренажер для повышения уровня сформированности компетенций в межкультурной коммуникации и развития лингвистической толерантности студентов вуза.
- Создание информационного дидактического ресурс, функционирующего в сети Интернет, с целью ознакомления, информирования об особенностях межнациональных контактов для повышения уровня толерантности и уровня безопасности межнациональных контактов на иностранном языке
- Оказание консультативной помощи в написании англоязычных статей сотрудникам НИУ «БелГУ».
- Установление, поддержание и расширение научных коммуникаций на общероссийском и международном уровне.
- Организация методической помощи по координации исследований в рамках своей и смежной тематики.
- Проведение научных конференций, семинаров и симпозиумов, организация временных научно-исследовательских коллективов по выполнению грантовых проектов с участием специалистов других вузов и научных центров.
- Организация подготовки и издания научных публикаций по результатам совместных исследований.
- Организация условий для проведения исследований отечественными и зарубежными специалистами.

1.4 Анализ технологий разработки

Для создания современных сайтов обычно используются контент-менеджер системы (CMS) или фреймворки, в основе которых лежат фундаментальные технологии: HTML, CSS, XML SQL и языки программирования, такие как: JavaScript PHP, Python, Ruby, Perl, C# и другие (Колисниченко, 2013).

1.4.1 Анализ языков программирования

Рассмотрим следующие языки программирования:

- PHP;
- Python;
- Ruby;
- ASP;
- JavaScript;
- Perl.

PHP - В основе лежит язык разметки HTML. PHP - это язык сценариев общего назначения, исходный код - открытый. Синтаксис достаточно легко поддается освоению, имеет немало общих черт с C, Java и Perl. Главное

преимущество PHP заключается в том, что с его помощью разработчики могут оперативно создавать динамически генерируемые веб-страницы. При профессиональном владении языком, его можно использовать и для выполнения других задач (Кауфман, 2016).

Python - В русском языке распространено как "питон". Высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций (Кауфман, 2016).

Ruby - В русском языке распространено как "руби". Динамический, рефлективный, интерпретируемый высокоуровневый язык программирования для быстрого и удобного объектно-ориентированного программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, сборщиком мусора и многими другими возможностями. По особенностям синтаксиса он близок к языкам Perl и Eiffel, по объектно-ориентированному подходу — к Smalltalk. Также некоторые черты языка взяты из Python (Кауфман, 2016).

ASP - Разработчиком данного языка является Microsoft. Технология позволяет разрабатывать приложения для WWW. ASP легко и быстро. Платформы для работы ASP: Windows NT и IIS (Internet Information Server). Не совсем корректно называть ASP языком, скорее, это именно технология для подключения программы к Web-страницам. Простой скриптовый язык и возможность использования внешних COM-компонентов - вот и весь секрет успеха ASP (Кауфман, 2016).

JavaScript - Принцип работы JavaScript несколько отличается от других языков программирование. Главное отличие состоит в том, что он подключается напрямую в HTML-файл. Сценарий, написанный на JavaScript,

проходит обработку интерпретатором, встроенным в браузер (Кауфман, 2016).

Области использования языка весьма обширны:

- создание веб-страниц, которые могут изменяться после загрузки документа;
- решение локальных задач;
- проверка грамотности заполнения форм пользователем до их пересылки на сервер.
- Многообразие возможностей javascript обуславливает популярность языка. С его помощью можно:
- вносить изменения на страницу: работать с тегами, менять стили, писать текст;
- реагировать на события (например, клик мыши) и выполнять определенную функцию;
- выводить сообщения, проверять корректность данных, устанавливать и считывать cookie;
- загружать данные без перезагрузки страницы и т.д.

Perl - Изначально этот язык был средством для соединения программок, выполняющих различные функции, в единый сценарий, позволяющий решить комплекс задач: обработка текста, администрирование и т.д. Сегодня Perl - это основное средство для создания приложений CGI. С его помощью выполняется администрирование веб-серверов и других систем. Простота и оперативность написания сценариев на данном языке привели к его адаптации на такие платформы, как Windows, Mac и т.д. Perl - открыт и доступен, исходные тексты интерпретатора можно получить совершенно бесплатно (Кауфман, 2016).

1.4.2 Анализ фреймворков для веб-разработки

Фреймворк представляет из себя некоторый программный каркас, реализующий базовую архитектуру проекта. Наиболее известными современными php-фреймворками является Zend Framework, Yii, CodeIgniter, Symfony и многие другие.

Фреймворки дают разработчикам очень мощный инструмент для разработки более гибких и менее подверженных ошибкам приложений в более короткий срок. Фреймворки часто помогают ускорить процесс разработки, обеспечивая необходимую функциональность. Они включают управление пользователями/правами, доступ к данным, кэширование и многое другое. Фреймворки помогают сфокусироваться на более важных деталях дизайна и легко управлять проектом.

Рассмотрим следующие современные PHP-фреймворки:

- Laravel;
- Zend Framework;
- CakePHP;
- Symfony;
- Yii;
- CodeIgniter.

Laravel - Один из самых популярных PHP-фреймворков, обладающий выразительным и элегантным синтаксисом. Он позволит максимально упростить решение основных и самых наболевших задач, таких как аутентификация, маршрутизация, сессии и кэширование. Laravel создавался, как попытка объединить только все лучшее, что есть в других PHP-

фреймворках, а также Ruby on Rails, ASP.NET MVC и Sinatra. Одно из самых важных его достоинств — наличие интегрированной системы модульного тестирования.

Zend Framework (ZF) - Написанный полностью в объектно-ориентированном стиле фреймворк, использующий все последние новшества PHP. Разработан с минимальными зависимостями от других компонентов, каждый из которых можно использовать отдельно; тем не менее, стандартный набор библиотек делает его очень мощным и легко расширяемым средством разработки. Кроме того, он предлагает надежную и высокопроизводительную реализацию MVC и удобную в использовании абстракцию базы данных, а также множество других возможностей, которые в сумме делают его одним из самых функциональных фреймворков.

CakePHP — написанный на PHP программный каркас для создания веб-приложений, обладающий активным и быстрорастущим сообществом. Как и большинство других фреймворков, реализует паттерн MVC. Изначально он создавался как клон популярного Ruby on Rails, и многие его идеи были заимствованы именно оттуда. От своих конкурентов отличается тем, что поддерживает не только PHP5, но и PHP4 (Воробьев, 2014).

Symfony – состоящий из множества компонентов и написанный на PHP5 фреймворк, использующий паттерн Model-View-Controller. Предлагает быструю разработку и управление веб-приложениями, позволяет легко решать рутинные задачи веб-программиста. Одно из основных его достоинств — поддержка множества баз данных (MySQL, PostgreSQL, SQLite или любая другая PDO-совместимая СУБД). Был выпущен под лицензией MIT. Symfony является свободным программным обеспечением. Symfony не следует путать с Symfony CMS, Open Source XML/XSLT системой управления конвентом. Symfony ускоряет создание и поддержку web-приложений. В настоящее время совместима с ORM Propel и Doctrine. Следует отметить, что на типичном хостинге, где ускоритель PHP

отсутствует, Symfony может использовать собственный движок кэширования для ускорения выполнения кода.

Symfony направлен на создание надежных приложений, с целью предоставить разработчикам полный контроль над конфигурацией.

Достоинства:

- простой в освоении;
- достаточно высокая скорость работы ядра.

Yii– высокопроизводительный фреймворк, использующий паттерн MVC и предназначенный для быстрой разработки современных web-приложений, изучение которого будет оправдано для организации, со штатом начиная от 10 человек, в одиночку с Yii справиться сложно. Фреймворк позволяет создавать очень производительные системы. В него уже заложено множество проверенных и готовых к использованию решений: конструктор запросов, ActiveRecord для реляционных и NoSQL баз данных, RESTful API, многоуровневая поддержка кэширования и многие другие.

Yii является золотой серединой между Zend Framework и Codeingiter. Он обладает великолепной документацией и всегда можно залезть в код, чтобы разобраться, что и как работает.

Фреймворк написан строго на PHP5 и строго в концепции ООП, поэтому если есть базовые знания принципов ООП, то разобраться в нем будет просто.

Yii намного более комплексный (больше функционала) и лучше структурирован, чем CodeIgniter. Он меньше и более производительный, чем Zend Framework. Требуется намного меньше кода при написании приложений, в то же время Yii обладает огромным функционалом(Воробьев, 2014).

CodeIgniter – Пожалуй, наиболее простой в освоении и использовании фреймворк. Легко расширяется, безопасен и использует простые и понятные

подходы, одним словом — идеален для новичков. Одно из других его преимуществ — скорость работы, этот фреймворк куда быстрее справляется с задачей работы с БД, чем другие его собратья. Имеет очень простой процесс установки, требующим минимальной конфигурации, так что начать будет просто. Он идеально работает практически на всех платформах виртуального и выделенного хостинга. CodeIgniter послужил базой для таких фреймворков, как Kohana и Rain Framework, многие идеи CodeIgniter применены во фреймворках Fuel PHP и CodeLighter. На CodeIgniter основано множество CMS: Fuel CMS, MaxSite CMS, Cogear, PyroCMS и другие. CodeIgniter - это PHP фреймворк, который использует платформу MVC, имеет классы для доступа к данным, классы для работы с почтой, FTP и XML-RPC. У CodeIgniter имеется исчерпывающая документация для начала работы (Воробьев, 2014).

1.4.3 Анализ шаблонизаторов php

Шаблонизатор должен способствовать выполнению концепции разделения. Язык шаблонизатора должен быть достаточно гибким и мощным чтобы упростить внедрение логики отображения, не дав возможности травмировать логику приложения расширенными возможностями.

Рассмотрим следующие шаблонизаторы:

- Smarty;
- Dwoo;
- Twig.

Smarty — компилирующий обработчик шаблонов для PHP, один из инструментов, позволяющих отделить прикладную логику и данные от представления в духе концепции model-view-controller.

Одно из предназначений Smarty — это отделение логики приложения от представления. Шаблоны, тем не менее, могут содержать в себе логику, но это должна быть логика представления данных. Она должна решать такие задачи, как подключение других шаблонов, чередующаяся окраска строчек в таблице, приведение букв к верхнему регистру, циклический проход по массиву для его отображения и т. п. Сама по себе библиотека Smarty не принуждает разделять логику приложения и представление — корректная дисциплина использования веб-шаблонов остаётся задачей разработчика.

Smarty позволяет:

- создавать пользовательские функции и модификаторы;
- использовать настраиваемые разделители тегов шаблона: {}, , и т. д.;
- возможность включения PHP-кода прямо в шаблон;
- пользовательские функции кэширования;
- использование компонентной архитектуры;
- объектно-ориентированная архитектура;
- автоматическое экранирование;
- наследование шаблонов.

Dwoo — это, прежде всего, интересный проект. Он позиционирует себя как альтернативу Smarty. Dwoo подражает Smarty, но также вносит и свои дополнения, например, наследование шаблонов, и работает гораздо быстрее Smarty.

Преимущества Dwoo:

- наследование шаблонов, подход без вложений;
- смартиподобный синтаксис с некоторыми модификациями;

- прекомпилируемые плагины могут быть как классами, так и просто функциями;
- наличие адаптеров для использования в ZF, CakePHP, Drupal CMF и т.п);
- поддержка Unicode / UTF-8.

Twig — компилирующий обработчик шаблонов с открытым исходным кодом, написанный на языке программирования PHP. Армин Ронахер написал Twig в 2008 году для платформы блогов Chyrp. Он больше не возвращался к разработке и в большей степени занимался разработкой на Python. Синтаксис языка шаблонов Twig берёт начало от движков шаблонов Jinja и Django, первый из которых также создан Ронахером. Идею данного шаблонизатора развивает и поддерживает Фабьен Потенсье, ведущий разработчик и идеолог фреймворка Symfony, в котором Twig используется по умолчанию.

Присутствует:

- встроенное наследование шаблонов (шаблоны компилируются как классы);
- автоматическое экранирование (отсутствует дополнительное время на запуск — все делается во время компиляции);
- очень безопасный режим «песочницы» (список допустимых тегов, фильтров и методов которые разрешены в шаблоне);
- расширяемость: вы можете переписывать все что угодно, даже функции ядра, написав расширение; так же можно манипулировать AST (Abstract Syntax Tree) перед компиляцией. Используя эти возможности, вы можете создать даже свой собственный язык — DSL (Domain Specific Language), ориентированный на ваше приложение.

1.4.4 Анализ СУБД

Рассмотрим следующие современные СУБД:

- MySQL;
- PostreSQL.

MySQL - является наиболее приспособленной, для применения в среде web, системой управления базами данных. Не секрет, что для исполнения приложений клиента на большинстве хостинг-площадок провайдеры предоставляют небольшое количество ресурсов. Поэтому для данного применения необходима высокоэффективная СУБД, обладающая высокой надежностью (Харрис, 2013).

Основные преимущества MySQL:

- многопоточность;
- оптимизация связей;
- записи фиксированной и переменной длины;
- ODBC драйвер;
- гибкая система привилегий и паролей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- быстрая работа;
- масштабируемость;
- совместимость с ANSI SQL;
- хорошая поддержка со стороны хостинг-провайдеров;
- быстрая поддержка транзакций.

PostreSQL - одна из наиболее серьезных СУБД.

Технические детали PostgreSQL:

- высокий уровень соответствия ANSI SQL 92, ANSI SQL 99 и ANSI SQL 2003, 2011;
- интерфейсы для Tcl, Perl, C, C++, PHP, JSON, ODBC, JDBC, Embedded SQL in C, Python, Ruby, Java и других;
- интеграция защиты данных с операционной системой (SE-Linux);
- представления, последовательности, наследование, outer joins, подзапросы, ссылочная целостность, оконные функции, CTE (рекурсивные запросы);
- пользовательские функции, хранимые процедуры, триггеры;
- процедурные языки PL/PgSQL, PL/Perl, PL/Python, PL/Java и другие;
- расширяемый набор типов данных с поддержкой индексов (GiST, GIN, SP-GiST);
- встроенная система полнотекстового поиска с поддержкой всех европейских языков;
- встроенная поддержка слабоструктурированных данных (xml, json, jsonb) с поддержкой индексов;
- горячее резервирование и репликация (синхронная, асинхронная, каскадная), PITR, двунаправленная (BDR);
- полная поддержка ACID, уровни изоляции, эффективная сериализация транзакций;
- функциональные и частичные индексы;
- интернационализация, поддержка Unicode и locale.

1.4.5 Анализ систем документирования исходных кодов

Документация кода отличается от проектной документации, так как она в основном фокусируется на том как работает система.

Основные причины документирования кода:

- Код будет поддерживаться и использоваться другими программистами в команде. Обслуживание кода становится большой проблемой, если он не был должным образом документирован.
- Документирование вашего кода делает логику гораздо более ясной, а также делает ваш код лучше.
- Недокументированный код тяжело не только передавать на сопровождение, а также порой тяжело сопровождать и самому.
- Рассмотрим следующие системы документирования исходного кода:
- Doxygen;
- PhpDocumentor.

Doxygen - это кроссплатформенная система документирования исходных текстов, которая поддерживает C++, Си, Objective-C, Python, Java, IDL, PHP, C#, Фортран, VHDL и, частично, D. Самым большим преимуществом использования Doxygen является то, что вы будете иметь последовательность всей документации исходного кода. Она также может помочь вам создавать структуру кода с использованием недокументированных исходных файлов. Все, что вам нужно сделать, это настроить его соответствующим образом.

PhpDocumentor - это система документирования исходных текстов на PHP. Имеет встроенную поддержку генерации документации в формате

HTML, LaTeX, man, RTF и XML. В основе работы системы лежит парсинг логической структуры PHP кода (классы, функции, переменные, константы) и привязка к ней комментариев, написанных по определенным стандартам. Инструмент также может помочь вам генерировать отчеты и графики и повысить общее качество кода. В ходе анализа также используется метаинформация об объектах программы, представленная в виде документирующих комментариев. На основе всей собранной информации формируется готовая документация, как правило, в одном из общепринятых форматов — HTML, HTMLHelp, PDF, RTF и других.

1.4.6 Анализ программных средств разработки

Для программирования нет необходимости в использовании мощного текстового процессора.

Рассмотрим текстовые редакторы Notepad++, Sublime Text и Atom.

Notepad++ - это бесплатный редактор текстовых файлов с поддержкой синтаксиса большого количества языков программирования. Программа располагает широким набором опций и отличается минимальным потреблением ресурсов процессора.

Основные достоинства Notepad++:

- авто-завершение набираемого слова;

- возможность создания собственного списка API функций (или скачать его со страницы загрузки);
- поддержка регулярных выражений Поиска/Замены;
- полная поддержка перетягивания фрагментов текста;
- динамическое изменение окон просмотра;
- автоматическое определение состояния файла (уведомление об изменении или удалении файла другой программой— с возможностью перезагрузить файл или удалить его из программы);
- увеличение и уменьшение (масштабирование);
- поддержка большого количества языков.

Среди продвинутых опций Notepad++ — опция подсветки текста и возможность сворачивания блоков, согласно синтаксису языка программирования. Пользователь может самостоятельно определить синтаксис языка программирования. Есть возможность настроить режим подсветки. Доступно выделение цветом директив и операторов языка программирования.

Notepad++ обеспечивает возможность одновременного просмотра и редактирования нескольких документов. Также Вы можете просматривать редактировать в двух окнах отображения один и тот же документ в разных местах. Изменение документа в одном окне просмотра будет автоматически перемещено во второе окно просмотра (т. е. Вы редактируете один документ, который имеет клона во втором окне просмотра).

Sublime Text - один из лучших текстовых редакторов на сегодняшний день. Это отличная альтернатива мощным IDE, он легки и делает свою работу с большой эффективностью, и точностью.

Преимущества:

- приятный, легкий, минималистичный интерфейс;
- очень гибко настраивается. Множественное выделение;

- возможность создания любых сниппетов и вставки их хоть по горячим клавишами, хоть по буквенным сокращениям (в стиле Zen Coding);
- возможность назначения горячих клавиш абсолютно на любое действие;
- в сниппетах можно задать, где будет находится курсор при вставке, задать плейсхолдеры и переключение в нужные участки сниппета Tab’ом;
- наличие миникарты кода для удобного перемещения;
- возможность отображения скрытых символов (пробелы, табы) только при выделении кода;
- есть множество доступных плагинов и постоянно растущее сообщество пользователей, которые пишут их под любые нужды.

Atom - как современный, гибко настраиваемый редактор с открытым исходным кодом. Несмотря на использование браузерных технологий, Atom не является web-приложением, а по сути, представляет собой специализированный вариант Chromium, в котором каждая вкладка выполняет роль локально обрабатываемой web-страницы.

Преимущества:

- Atom является редактором с открытым исходным кодом, который свободен в использовании;
- кроссплатформенность OS X, Windows и Linux;
- умное автодополнение;
- браузер файлов;
- поиск и замена по многим файлам;
- прост в использовании даже для новичка;
- множество дополнительных плагинов.

Вывод к Главе 1

На основании рассмотренного выше материала стала понятна необходимость создания веб-ресурса для НИЛЯМ и для реализации проекта были выбраны следующие технологии:

PHP - за распространённость языка, его синтаксис достаточно легко поддается освоению, оперативное создание динамически генерируемых веб-страниц

CodeIgniter — за простоту в освоении; использование простых и понятных подходов; простота установки и настройки; идеально работает практически на всех платформах виртуального и выделенного хостинга.

Smarty - наследование шаблонов, позволяющее ускорить процесс разработки, а также за использование компонентной архитектуры, которая избавляет от повторяемого кода.

MySQL — популярная и очень распространенная СУБД среди хостинг-провайдеров.

Doxygen - простота использования утилиты; освоение синтаксиса позволит в дальнейшем документировать код и для текстов других языков программирования.

Atom - прост в настройке, есть возможность установить дополнительные плагины; приятный пользовательский интерфейс.

Глава 2 Разработка сайта НИЛЯМ

2.1 Знакомство с архитектурой MVC (Model-View-Controller)

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то как они отображаются, не трогая саму Модель.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели. Однако, представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

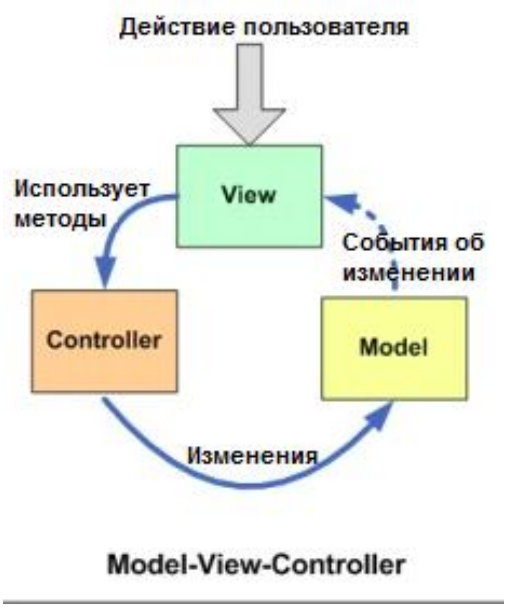

Рис. 2.1. Взаимодействие пользователя с приложением построенным на MVC-архитектуре

Основная идея этого паттерна в том, что и контроллер, и представление зависят от модели, но модель никак не зависит от этих двух компонент.

2.2 Проектирование БД

Для проектирования БД воспользуемся веб-интерфейсом предоставляемым.

ПО PhpMyAdmin.

Итак, создадим таблицы под хранение следующих данных:

- данные пользователя;
- данные статьи;
- данные под дополнительные ресурсы стать.

Таблица для хранения данных пользователей - «user».

Поля таблицы:

- name - ФИО пользователя;
- email - Email пользователя, в данном случае используется как логин для входа в личный кабинет на сайте;
- password - Пароль пользователя, также используется для входа в личный кабинет на сайте;
- restore_password - Значение нового пароль для восстановления, заменяет значение поля password при подтверждении восстановления;
- registration_at - Дата регистрации;
- lastlogin_at - Дата последнего входа в личный кабинет;
- is_admin - Логический признак администратора;
- is_active - Логический признак доступа пользователя к функционалу добавления и редактирования собственных статей;
- about_me - Подробная информация о авторе статьи, необязательное поле для заполнения из профиля;
- avatar - Фотография пользователя, необязательное поле для заполнения из профиля;
- work_position - Должность автора статьи, необязательное поле для заполнения из профиля.

Таблица для хранения основных данных статьи - «article».

Поля таблицы:

- title - Заголовок статьи;
- author_id - Идентификатор автора статьи (значение поля id таблицы user);
- content - Основной текст статьи;
- keywords - Ключевые слова статьи;
- description - Краткое описание статьи;
- create_at - Дата создания;

– modified_at - Дата внесения последних изменений;
– is_active - Признак статьи в открытом доступе.

Таблица для хранения дополнительных материалов статьи - «article_material».

Поля таблицы:

– article_id - Числовой идентификатор материала;
– title - Заголовок материала;
– val - Значение материала, наименование файла;
– sort - Числовое значение приоритета сортировки;
– is_image - Логический признак изображения материала;
– file_ext - Расширение файла.

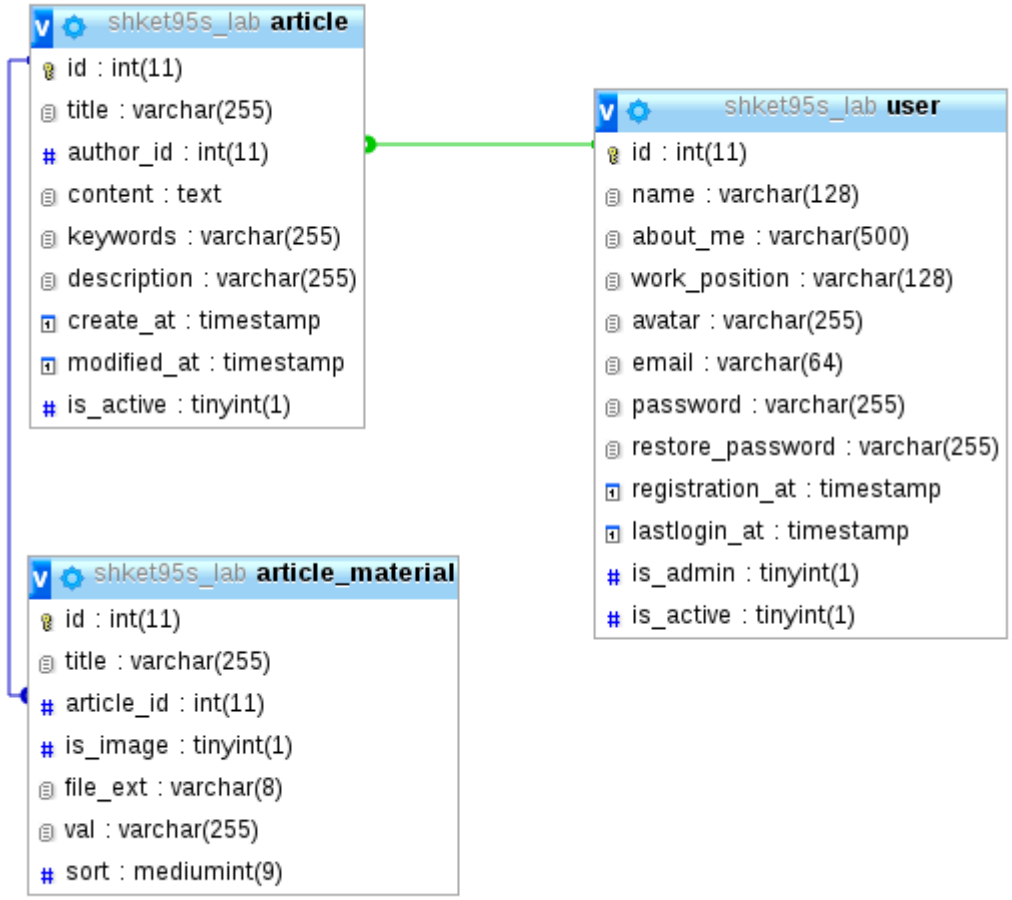

Рис. 2.2. Вид таблиц спроектированной БД в дизайнере PhpMyAdmin

2.3 Программирование моделей

Основываясь на таблицах БД и каркаса фреймворка CodeIgniter — реализуем классы моделей.

Класс модели наследуется от базового класса фреймворка CI_Model. Для каждой таблицы БД создаётся отдельная модель с определением полей таблиц и методов ввода/вывода данных из них.

Классы моделей:

- User — Класс модели таблицы пользователей;
- Article — Класс модели таблицы статей;
- ArticleMaterial — Класс модели таблицы дополнительных материалов статей.

Обязательным условием реализации является вызов конструктора родительского класса в конструкторе класса наследуемой модели:

```
public function __construct()
{
        parent::__construct();
}.
```

2.4 Программирование представления (шаблонов)

Следуя принципам архитектуры MVC, создадим шаблоны страниц и их составных частей.

Для создания шаблонов воспользуемся шаблонизатором Smarty. Файлы шаблонов создаются в папке application/views/templates. Для шаблонизатора Smarty, файлам шаблонов дописываем расширение файлов - .tpl (от слова template).

Разделим шаблоны на 4-е группы:

- общие шаблоны
- шаблоны статьи
- шаблоны в личном кабинете редактора
- шаблоны в административной части.

Общие шаблоны.

Общие шаблоны находятся в папке приложения: /application/views/templates.

Перечень шаблонов см. в табл. 2.1.

Таблица 2.1. Общие шаблоны

Наименование файла	**Назначение**
base	Базовый шаблон, от которого наследуются все остальные шаблоны
header	Заголовочная информация сайта в котором присутствую блоки для вставки значений заголовка страницы, ключевых слов и краткого описания. Шаблон наследуется всеми основными страницами сайта.
footer	Шаблон «подвала» страницы, где располагается текст копирайта.
menu	Меню сайта
pagination	Пагинатор страниц
index	Главная страница сайта с текстом о лаборатории
info	Страница «Информация»
contacts	Страница «Контакты»

signup	Страница «Регистрация»
signin	Страница «Авторизация»
restore_password	Страница «Восстановление пароля»

Шаблоны статьи.

Шаблоны статей находятся в папке приложения: /application/views/templates/article.

Перечень шаблонов см. в табл. 2.2.

Таблица 2.2. Шаблоны статьи

Наименование файла	**Назначение**
detail	Вывода детального описания статьи
form	Форма добавления/редактирования статьи
list	Список статей, используется при переходе из пункта меню «Статьи»
succesful_removed	Сообщение об успешном удалении статьи
error_removed	Сообщение об ошибке удаления статьи

Шаблоны личного кабинета пользователя-редактора.

Шаблоны находятся в папке приложения: /application/views/templates/author.

Перечень шаблонов см. в табл. 2.3.

Таблица 2.3. Шаблоны личного кабинета

Наименование файла	**Назначение**
article_list	Список статей пользователя
profile	Форма профиля пользователя
detail	Вывод детальной информации автора

list	Список активных авторов статей, используется при переходе из пункта меню «Авторы»
info	Сообщение о неактивности личного кабинета (активация производится администратором сайта)
successful_restore_password	Сообщение об успешной отправке письма на почту пользователя для восстановления пароля

Шаблоны в административной части.

Шаблоны располагаются в папке приложения: /application/views/templates/admin

Перечень шаблонов см. в табл. 2.4.

Таблица 2.4. Шаблоны в административной части

Наименование файла	**Назначение**
article_list	Список статей всех авторов
editor_list	Список всех авторов

2.5 Программирование контроллеров и задание маршрутов действий

Основываясь на каркасе фреймворка CodeIgniter и архитектуре MVC— реализуем классы контроллеров.

Класс контроллера наследуется от базового класса фреймворка CI_Controller.

Реализуем 4-е контроллера:

- Контроллер для всех ролей пользователей (гостей, редакторов и администратора) - «Clab».
- Контроллер администратора - «ClabAdmin».
- Контроллер редактора, автора статей - «ClabEditor».
- Контроллер общий для администратора и редактора, который предоставляет доступ к функционалу публикации статей - «CLabArticle».

Используя файла настроек машрутов /application/config/routes.php определим следующие маршруты и действия для наших контроллеров см. в табл. 2.5.

Таблица 2.5. Соответствия маршрута и действия

Маршрут	**Котроллер/Действие**	**Назначение**
default_controller	CLab	Точка входа на сайт. Главная страница.
/quit	CLab/quit	Выход из личного кабинета
/articles /articles/(:num)	CLab/articles	Статьи
/article/(:num)	ClabArticle/detail/	Детальный просмотр статьи
/info	CLab/info	Страница «Информация»
/authors	CLab/authors	Страница «Авторы»
/contacts	CLab/contacts	Страница «Контакты»

/signin	CLab/signin	Страница «Авторизация»
/signup	CLab/signup	Страница «Регистрация»
/restore_password	CLab/restore_password	Страница «Восстановление пароля»
/successful_restore_password	CLab/successful_restore_password	Страница сообщения о успешной отправке нового пароля на эл. почтовый ящик
/confirm_restore_password/(:num)/(:any)	CLab/confirm_restore_password/$1/$2	Страница подтверждения изменения нового пароля
/authors /authors/(:num)	CLab/authors	Страница «Авторы»
/author/(:num)	CLabEditor/detail	Страница детального просмотра информации об авторе
/editor/articles /editor/articles/(:num)	CLabEditor/articles	Список статей редактора
/editor/article/edit/(:num)	CLabArticle/edit/$1	Редактирование статьи
/editor/article/add	CLabArticle/add	Добавление статьи
/editor/article/remove/(:num)	CLabArticle/remove/$1	Удаление статьи
/admin/articles /admin/articles/(:num)	CLabAdmin/articles	Список всех статей
/admin/editors /admin/editors/(:num)	CLabAdmin/editors	Список зарегистрированных авторов

/admin/editors/status/(:num)/(:num)	CLabAdmin/active_status/	Изменение признака активности автора

В скобках :num и :any обозначаются типы параметров — числовой и любой, соответственно.

2.6 Создание документации исходного кода проекта

Для создания документации кода нашего проекта мы воспользуемся системой документирования исходных текстов — Doxygen.

Doxygen генерирует документацию на основе набора исходных текстов и также может быть настроен для извлечения структуры программы из недокументированных исходных кодов. Возможно составление графов зависимостей программных объектов, диаграмм классов и исходных кодов с гиперссылками.

Для начала создадим файл настроек для системы документирования командой:

doxygen -g <config_name>.

Таблица 2.6. Основные опции

Тэг	**Назначение**	**Значение по умолчанию**
DOXYFILE_ENCODING	Кодировка, которая используется для всех символов в данном	UTF-8

	файле настроек	
OUTPUT_LANGUAGE	Устанавливает язык, на котором будет сгенерирована документация	English
PROJECT_NAME	Название проекта, которое может представлять собой единое слово или последовательность слов (если вы редактируете вне Doxywizard, последовательность слов необходимо поместить в двойные кавычки)	My Project
PROJECT_NUMBER	Данный тэг может быть использован для указания номера проекта или его версии	—
PROJECT_BRIEF	Краткое однострочное описание проекта, которое размещается сверху каждой страницы и даёт общее представление о назначении проекта	—
OUTPUT_DIRECTORY	Абсолютный или относительный путь, по которому будет сгенерирована документация	Текущая директория
INPUT	Список файлов и/или директорий, разделенных пробелом, которые содержат в себе исходные коды проекта	Текущая директория
RECURSIVE	Используется в том	NO

	случае, если необходимо сканировать исходные коды в подпапках указанных директорий	
INLINE_SOURCES	Вставка кода тел методов и функций	NO
PROJECT_LOGO	Путь к изображению логотипа	-
HTML_COLORSTYLE_HUE	Указывает тон цветовой схемы	220
HTML_COLORSTYLE_SAT	Указывает яркость цветовой схемы	100
HTML_COLORSTYLE_GAMMA	Указывает гамму цветовой схемы	80
DISABLE_INDEX	Отключает отображение меню с вкладками	NO
GENERATE_TREEVIEW	Показывает отображать или нет иерархическое дерево	NO
HTML_DYNAMIC_SECTIONS	В случае установки этой опции, Doxygen будет сворачивать определенные элементы HTML документации (например, графы), которые пользователь по желанию может впоследствии раскрыть	NO

Запуск создания документации:

doxygen <config_file>.

2.7 Администрирование статей и пользователей

Для публикации статей, автору статей необходимо иметь учётную запись на сайте (быть зарегистрированным пользователем).

Перечень раздела руководства:

- Регистрация на сайте
- Авторизация
- Добавление статьи
- Редактирование статьи
- Заполнение профиля пользователя
- Администрирование статей и пользователей

Для того, чтобы зарегистрироваться на сайте :

1) Нажмите на кнопку в меню сайта «Регистрация»

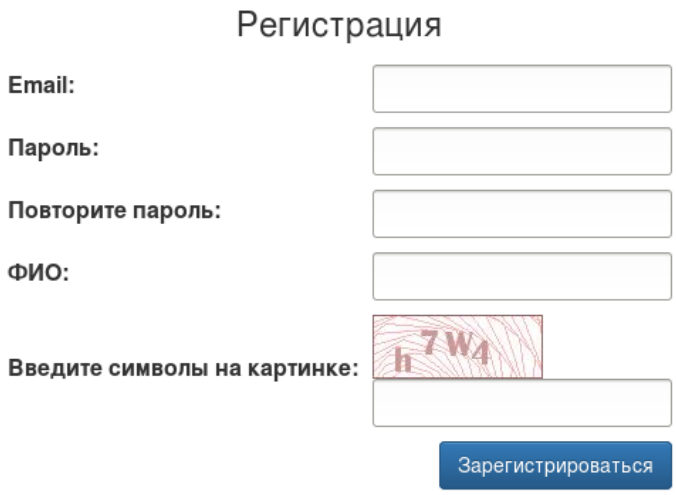

Рис. 2.3. Форма регистрации

2) Заполните все поля формы регистрации.
3) Нажмите на кнопку «Зарегистрироваться»

4) После регистрации, Вы будете автоматически направлены на страницу своего профиля, который заполняете.
5) Выйдите из личного кабинета нажав на кнопку «Выйти» в меню сайта.
6) Свяжитесь администратором сайта для активации Вашей учётной записи.

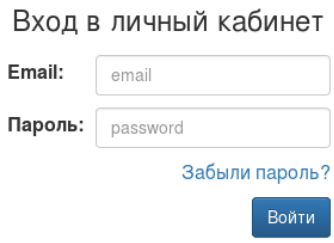

Рис. 2.4. Форма авторизации

После регистрации и активации вашей учётной записи, перейдите на страницу формы авторизации, нажав на кнопку «Войти» в меню сайта. Заполните поля формы и нажмите на кнопку «Войти», после чего Вы будете автоматически перенаправлены на страницу Вашего профиля.

Для добавления элементов на сайт:

1) Перейдите на страницу списка статей, выбрав из меню «Мой кабинет» пункт «Мои статьи»

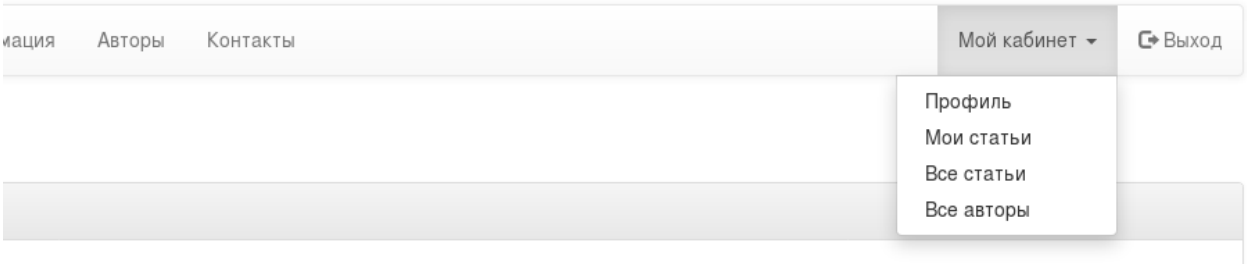

Рис. 2.5. Меню пункта «Мой кабинет» для администратора

2) После чего загрузится страница со списком Ваших статей

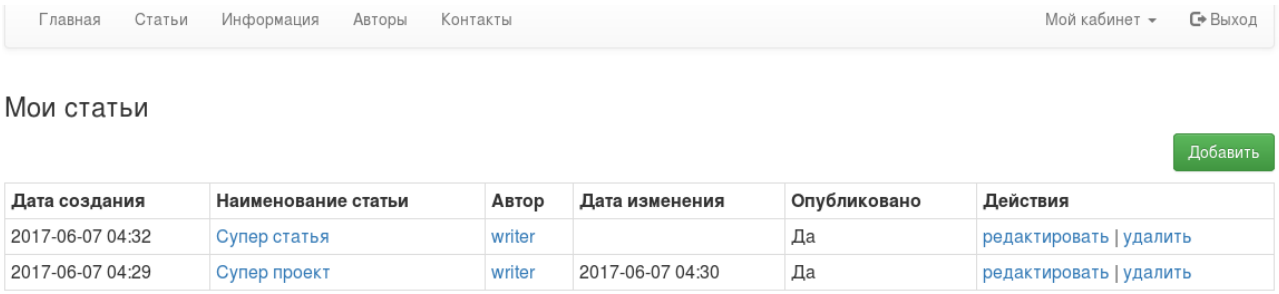

Рис. 2.6. Страница списка статей пользователя

3) Нажмите на зелёную кнопку «Добавить»
4) После чего загрузиться страница с формой добавления статьи
5) В конце редактирования статьи необходимо нажать на кнопку «Сохранить», чтобы сохранить статью в БД. После чего пользователь будет перенаправлен на страницу списка своих статей.

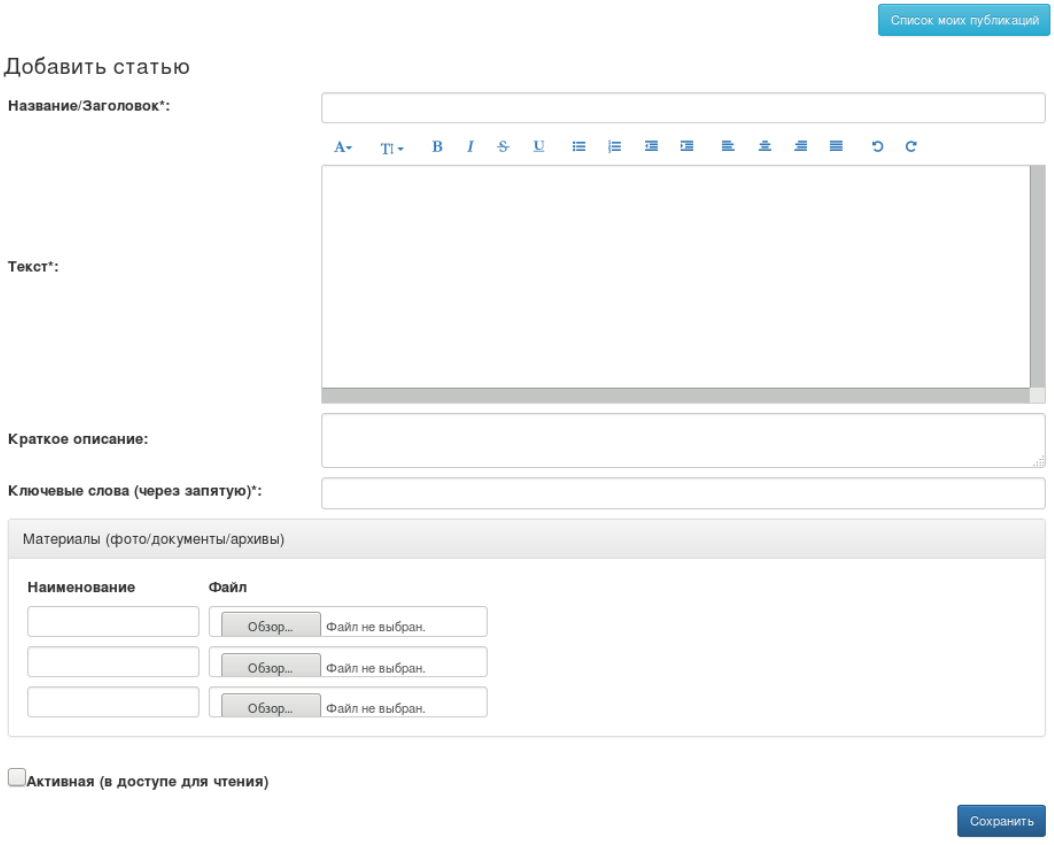

Рис. 2.7. Форма добавления/редактирования статьи

Для того, чтобы отредактировать статьи:

1) Перейдите на страницу «Мои статьи»

2) Напротив статьи в таблице нажмите ссылку «редактировать»
3) Откроется форма редактирования статьи
4) После правок, нажмите на кнопку «Сохранить»

Для того, чтобы продолжить работу необходимо авторизоваться на сайте. После авторизации Вы окажетесь на странице Вашего профиля. После заполнения полей необходимо нажать кнопку «Сохранить».

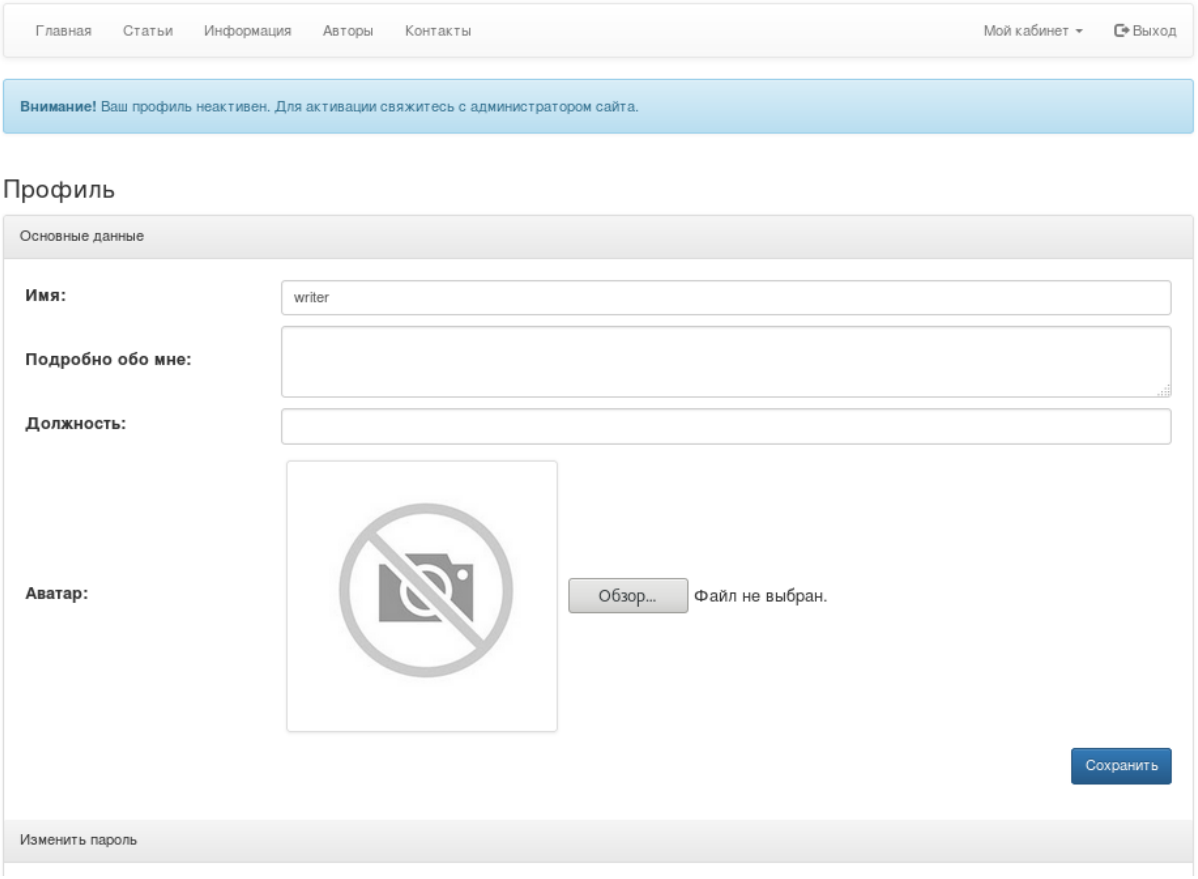

Рис. 2.8. Вид страницы профиля после регистрации

Администрировать статьи и пользователей имеет право администратор сайта — первый зарегистрированный пользователь.

На странице «Все статьи», администратор может удалять статьи других пользователей.

На странице «Все авторы», администратор может активировать или деактивировать любого пользователя. При деактивации автора, автор лишается возможности добавлять и редактировать свои статьи — может только удалять.

Вывод к Главе 2

На основании проведенного анализа деятельности Научно-исследовательской лаборатории языкового мониторинга было установлено, что лаборатория нуждается в создании web-ресурса. Учитывая это, был разработан web-сайт, презентующий НИЛЯМ.

Существенно помог и ускорил процесс разработки шаблонизатор Smarty, который предоставляет возможность наследовать другие шаблоны, а также вставлять их код в любое место кода другого шаблона, благодаря чему количество кода меньше и он лучше структурирован. Также, синтаксис шаблонизатора улучшил читаемость кода шаблона, т.к. избавил нас от прямой вставки php-кода с командами вывода информации - это также очень поспособствовало ускорению написания сайта.

Отдельно стоит отметить сам фреймворк CodeIgniter. Используемая фреймворком парадигма MVC (модель-представление-контроллер) позволяет не только хорошо структурировать код, но и легко расширять функционал проекта при необходимости. Как и многие другие фреймворки, он предоставляет разработчику готовые решения тривиальных задач, таких как: URL-маршрутизация, получение и передача параметров, связь с БД и простую работу с ней с использованием простого ORM (объектно-реляционного отображения), что всё в целом позволяет разработчику сосредоточиться непосредственно на реализации поставленных перед ним задач.

ЗАКЛЮЧЕНИЕ

В дипломной работе были подробно рассмотрены особенности и этапы создания web-сайта на основе фреймворка и шаблонизатора. И на основе представленного теоретического материала был разработан сайт, полностью готовый к применению.

При разработке web-ресурса были проанализированы новейшие технологии, которые позволяют создавать интерактивные web-страницы. Самыми подходящими для реализации поставленной задачи оказались фреймворк CodeIgniter и шаблонизатор Smarty.

Основным результатом дипломной работы является создание полнофункционального web-сайта для Научно-исследовательской лаборатории языкового мониторинга, чья деятельность заключается в исследовании вариативности английского языка Великобритании в условиях влияния различных факторов. С помощью сайта пользователи смогут получать необходимую информацию по работе научной лаборатории, связаться с работниками и задать интересующие вопросы, ознакомиться с опубликованными научными статьями.

Отсюда следует, что поставленные цели и задачи данной работы выполнены в полной мере.

На основе исходных текстов с комментариями, с помощью системы документирования исходных текстов была сгенерирована документация для поддержки проекта любым разработчиком. Это дает возможность усовершенствовать сайт с целью дальнейшего повышения его информативности, привлекательности и удобства, позволяет расширить и развить проект в дальнейшем.

СПИСОК ИСПОЛЬЗУЕМЫХ РЕСУРСОВ

1. Bootstrap 3 Tutorial [Электронный ресурс]. – 2017. - Режим доступа: https://www.w3schools.com/bootstrap/.
2. CodeIgniter + Smarty Perfect Together [Электронный ресурс]. – 2017. - Режим доступа http://www.coolphptools.com/codeigniter-smarty..
3. Doxygen [Электронный ресурс]. – 2017. - Режим доступа: https://ru.wikipedia.org/wiki/Doxygen.
4. Getting Started with Bower [Электронный ресурс]. – 2017. - Режим доступа: http://blog.teamtreehouse.com/getting-started-bower.
5. Model-View-Controller [Электронный ресурс]. – 2017. - Режим доступа: https://ru.wikipedia.org/wiki/Model-View-Controller.
6. Алан Б. ИЗУЧАЕМ SQL. – Символ®; ББК: 3973. 26-018.19 SQL, 0 УДК: 004.438, 2013.
7. Бенкен Е. С. PHP, MySQL, XML: программирование для Интернета, 2 изд. – БХВ-Петербург, 2012.
8. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL //М.: Изд. дом «Вильямс. – 2012.
9. Воробьев В. В. Сравнение PHP фреймворков: Yii, CakePHP, CodeIgniter //Молодежный научно-технический вестник. – 2014. – №. 6. – С. 21-21.
10. Гончаров А. самоучитель HTML. – Издательский дом «Питер», 2012.
11. Гутманс Э., Баккен С., Ретанс Д. PHP 5. Профессиональное программирование //Пер. с англ.–СПб: Символ плюс. – 2014. – Т. 704.
12. Документируем код эффективно при помощи Doxygen [Электронный ресурс]. – 2017. - Режим доступа: https://habrahabr.ru/post/252101/.
13. Дронов В. А. JavaScript и AJAX в Web-дизайне. – БХВ-Петербург, 2012.

14. Кауфман В. Языки программирования. Концепции и принципы. – Litres, 2016.
15. Климов А. П. JavaScript на примерах, 2 изд. – БХВ-Петербург, 2012.
16. Колисниченко Д. Н. PHP и MySQL. Разработка Web-приложений. 4-е изд. – БХВ-Петербург, 2013.
17. Колисниченко Д. Н. Движок для вашего сайта. CMS Joomla!, Slaed, PHP-Nuke. – БХВ-Петербург, 2012.
18. Кузнецов М. В. PHP 5. Практика создания web-сайтов (+ CD). – БХВ-Петербург, 2013.
19. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS. 2-е изд //Никсон–СПБ.: Питер. – 2013.
20. Паттерны для новичков: MVC vs MVP vs MVVM [Электронный ресурс]. – 2017. - Режим доступа: https://habrahabr.ru/post/215605/ .
21. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера.(+ фтп) 4-е изд. – БХВ-Петербург, 2015.
22. Руководство по Smarty [Электронный ресурс]. – 2017. - Режим доступа: http://www.smarty.net/docsv2/ru/.
23. Руководство Пользователя CodeIgniter [Электронный ресурс]. – 2017. - Режим доступа: http://codeigniter3.info/.
24. Ташков П. А., Ташков П. А. Веб-мастеринг: HTML, CSS, JavaScript, PHP, CMS, графика, раскрутка: Html, Css, Javascript, Php, Cms, Grafika, Raskrutka. – Издательский дом" Питер", 2015.
25. Харрис Э. PHP/MySQL для начинающих. – М. : ИД Кудиц-Образ, 2013.

ПРИЛОЖЕНИЕ

```
<?php
/**
	\brief Класс модели статьи.
	Данный класс описывает ввод/вывод данных таблицы article БД
*/
class Article extends CI_Model
{

	public $title; ///< Заголовок статьи
	public $author_id; ///< Идентификатор автора статьи (значение поля id таблицы user)
	public $content; ///< Основной текст статьи
	public $keywords; ///< Ключевые слова статьи
	public $description; ///< Краткое описание статьи
	public $create_at; ///< Дата создания
	public $modified_at; ///< Дата внесения последних изменений
	public $is_active; ///< Признак статьи в открытом доступе

	/**
		\brief Конструктор класса, который в теле вызывает конструктор родительского класса.
		*/
		public function __construct()
		{
			parent::__construct();
		}

		/**
		\brief Метод добавления/вставки новых данных в таблицу
		\param[in] $arr_fields ассоциативный массив значений, в котором ключами являются наименование полей таблицы
		\return Числовой идентификатор записи в таблице
		*/
		function add($arr_fields)
		{
			foreach($arr_fields as $key => $val)
				$this->{$key} = $val;
			$this->db->insert('article', $this);
```

```
            return $this->db->query('SELECT LAST_INSERT_ID() as id')->result_array()[0]['id'];
        }
        /**
        \brief Метод обновления данных
        \param[in] $id Числовой идентификатор статьи в таблице article
        \param[in] $arr_fields Ассоциативный массив значений, в котором ключами являются наименование полей таблицы
        \return Логический признак успешного обновления (true или false)
        */
        function edit($id, $arr_fields)
        {
            foreach($arr_fields as $key => $val)
                $this->{$key} = $val;
            return  $this->db->update('article', $this, array('id' => $id));
        }
        /**
        \brief Метод  все данные по статье
        \param[in] $id Числовой идентификатор статьи в таблице article
        \return Ассоциативный массив значений, в котором ключами являются наименование полей описанных в sql-запросе
        Ключи:
            id, title, content, keywords, description, create_at, modified_at, is_active, author_name, materials
        где:
            ключ author_id - числовой идентификатор автора в таблице user
            ключ author_name - наименование автора
            ключ materials представлен ввиде JSON строки. Значения после преобразования извлекаются по следующим ключам:
            s - порядок сортировки;
            t - заголовок сортировки
            i - логический признак изображения материала
            val - значение материала, путь к файлу
            id - числовой идентификатор материала в таблице article_material
        */
        function get_detail($id)
        {
            $sql = "
                SELECT
                a.id,
                a.title,
                a.content,
                a.keywords,
                a.description,
```

```
                a.author_id,
                a.create_at,
                a.modified_at,
                a.is_active,
                u.name AS
author_name,

        GROUP_CONCAT(DISTINCT '{',
'\"s\":\"', am.sort, '\",', '\"t\":\"', am.title, '\",',
'\"i\":\"', am.is_image, '\",',  '\"val\":\"',
am.val, am.file_ext, '\",', '\"id\":', am.id, '}')
AS materials
                FROM article AS a
                LEFT JOIN user AS u
ON a.author_id = u.id
                LEFT JOIN
article_material AS am ON a.id =
am.article_id
                WHERE
                a.id = ?
                GROUP BY a.id
            ";
            return $this->db->query($sql,
array($id))->result_array()[0];
        }
        /**
        \brief Метод удаления записи
статьи из таблицы article
        \param[in] $id Числовой
идентификатор статьи в таблице article
        \return Логический признак
успешного удаления (true или false)
        */
        function remove($id)
        {
            return $this->db->where("id",
$id)->delete('article');
        }
        /**
        \brief Метод удаления записей
статей по автору
        \param[in] $author_id Числовой
идентификатор автора в таблице user
        \return Логический признак
успешного удаления (true или false)
        */
        function
remove_by_author($author_id)
        {
            return $this->db-
>where("author_id", $author_id)-
>delete('article');
        }
        /**
        \brief Метод  список статей по
автору
        \param[in] $author_id Числовой
идентификатор автора в таблице user
        \param[in] $per_page Кол-во
записей статей на страницу, по
умолчанию 10
        \param[in] $num_page Номер
страницы, по умолчанию 1
```

```
        \return Ассоциативный массив значений, в котором ключами rows и total_rows, где:
                ключ rows - это ассоциативный массив значений, где ключи являются наименование полей описанных в sql-запросе
                        (id, title, content, keywords, description, create_at, modified_at, is_active, author_id, author_name).
                ключи total_rows - общее кол-во записей (статей) автора
        */
        function get_list_by_author($author_id, $per_page = 10, $num_page = 1)
        {
                $offset = ($per_page * $num_page) - $per_page;
                $sql_count = "
                        SELECT COUNT(a.id) as quantity
                        FROM article a
                        WHERE a.author_id = ?
                ";
                $sql = "
                SELECT
                a.id,
                a.title,
                a.content,
                a.description,
                a.author_id,
                a.create_at,
                a.modified_at,
                a.is_active,
                u.name AS author_name
                FROM article AS a
                LEFT JOIN user AS u ON a.author_id = u.id
                WHERE a.author_id = ?
                GROUP BY a.id
                ORDER BY a.create_at DESC, a.id DESC
                LIMIT ? OFFSET ?
                ";
                $q = $this->db->query($sql, array($author_id, $per_page, $offset));
                $qcount = $this->db->query($sql_count, array($author_id));
                $ret = array();
                $ret['rows'] = $q->result_array();
                $r = $qcount->result_array();
                $ret['total_rows'] = $r[0]['quantity'];
                return $ret;
        }
        /**
        \brief Метод  список статей
```

```
        \param[in] $per_page Кол-во записей статей на страницу, по умолчанию 10

        \param[in] $num_page Номер страницы, по умолчанию 1

        \return Ассоциативный массив значений, в котором ключами rows и total_rows, где:

                ключ rows - это массив с ассоциативными массивами значений, где ключи являются наименование полей описанных в sql-запросе

                        (id, title, content, keywords, description, create_at, modified_at, is_active, author_id, author_name).

                ключи total_rows - общее кол-во записей (статей)

        */

        function get_list($per_page = 10, $num_page = 1)

        {

                $offset = ($per_page * $num_page) - $per_page;

                $sql_count = "

                        SELECT COUNT(a.id) as quantity

                        FROM article a

                        WHERE a.is_active = 1

                ";

                $sql = "

                SELECT

                        a.id,

                        a.title,

                        a.content,

                        a.description,

                        a.author_id,

                        a.create_at,

                        a.modified_at,

                        a.is_active,

                        u.name AS author_name

                        FROM article AS a

                        LEFT JOIN user AS u ON a.author_id = u.id

                        WHERE a.is_active = 1

                        GROUP BY a.id

                        ORDER BY a.create_at DESC, a.id DESC

                        LIMIT ? OFFSET ?

                        ";

                $q = $this->db->query($sql, array($per_page, $offset));

                $qcount = $this->db->query($sql_count, array());

                $ret = array();

                $ret['rows'] = $q->result_array();

                $r = $qcount->result_array();

                $ret['total_rows'] = $r[0]['quantity'];

                return $ret;
```

```
    }

    /**

    \brief Метод  список статей. Используется для вывода в административном разделе сайта.

    \param[in] $per_page Кол-во записей статей на страницу, по умолчанию 10

    \param[in] $num_page Номер страницы, по умолчанию 1

    \return Ассоциативный массив значений, в котором ключами rows и total_rows, где:

        ключ rows - это массив с ассоциативными массивами значений, где ключи являются наименование полей описанных в sql-запросе

            (id, title, content, keywords, description, create_at, modified_at, is_active, author_id, author_name).

        ключи total_rows - общее кол-во записей (статей)

    */

    function get_list_for_admin($per_page = 20, $num_page = 1)

    {

        $offset = ($per_page * $num_page) - $per_page;

        $sql_count = "

            SELECT COUNT(a.id) as quantity

            FROM article a

        ";

        $sql = "

        SELECT

        a.id,

        a.title,

        a.content,

        a.description,

        a.author_id,

        a.create_at,

        a.modified_at,

        a.is_active,

        u.name AS author_name

        FROM article AS a

        LEFT JOIN user AS u ON a.author_id = u.id

        GROUP BY a.id

        ORDER BY a.create_at DESC, a.id DESC

        LIMIT ? OFFSET ?

        ";

        $q = $this->db->query($sql, array($per_page, $offset));

        $qcount = $this->db->query($sql_count, array());

        $ret = array();

        $ret['rows'] = $q->result_array();

        $r = $qcount->result_array();
```

```
		$ret['total_rows'] = $r[0]['quantity'];
		return $ret;
	}
	/**
	\brief Метод  числовой идентификатор автора статьи.
	\param[in] $article_id числовой идентификатор статьи в таблице article
	\return числовой идентификатор автора в таблице user
	*/
	function get_author($article_id)
	{
		return $this->db->select('author_id')->where('id', $article_id)->get('article')->result_array()[0]['author_id'];
	}
	/**
	\brief Метод очищает таблицу article от всех записей (удаляет все записи таблицы).
	*/
	function clear()
	{
		$this->db->empty_table('article');
	}
}

<?php
/**
	\brief Класс модели дополнительных материалов статей.
	Данный класс описывает ввод/вывод данных таблицы article_material БД
*/
class ArticleMaterial extends CI_Model
{
	public $article_id; ///< Числовой идентификатор материала
	public $title; ///< Заголовок материала
	public $val; ///< Значение материала, наименование файла
	public $sort; ///< Числовое значение приоритета сортировки
	public $is_image; ///< Логический признак изображения материала
	public $file_ext; ///< Расширение файла
	/**
	\brief Конструктор класса, который в теле вызывает конструктор родительского класса.
	*/
	public function __construct()
	{
		parent::__construct();
	}
	/**
```

```
        \brief Метод добавления/вставки новых данных в таблицу

        \param[in] $arr_fields ассоциативный массив значений, в котором ключами являются наименование полей таблицы

        \return Числовой идентификатор записи в таблице

        */

        function add($arr_fields)

        {

                foreach($arr_fields as $key => $val)

                        $this->{$key} = $val;

                $this->db->insert('article_material', $this);

                return $this->db->query('SELECT LAST_INSERT_ID() as id')->result_array()[0]['id'];

        }

        /**

        \brief Метод удаления записи материала из таблицы article_material

        \param[in] $id Числовой идентификатор материала в таблице article_material

        \return Логический признак успешного удаления (true или false)

        */

        function remove($id)

        {

                return $this->db->where("id", $id)->delete('article_material');

        }

        /**

        \brief Метод удаления записей материала конкретной статьи

        \param[in] $article_id Числовой идентификатор статьи в таблице article_material

        \return Логический признак успешного удаления (true или false)

        */

        function remove_by_article($article_id)

        {

                return $this->db->where("article_id", $article_id)->delete('article_material');

        }

        /**

        \brief Метод возвращает список материалов конкретной статьи

        \param[in] $article_id Числовой идентификатор статьи в таблице article_material

        \return Массив с ассоциативными массивами значений, с ключами наименований полей таблицы article_material

        */

        function get_list($article_id)

        {
```

```
		return $this->db->where('article_id', $article_id)->order_by('sort', 'asc')->get('article_material')->result_array();
	}
	/**
	\brief Метод возвращает список материалов группы идентификаторов материалов
	\param[in] $arr_ids Массив числовых идентификаторов материалов в таблице article_material
	\return Массив с ассоциативными массивами значений, с ключами наименований полей таблицы article_material
	*/
	function get_list_by_arr($arr_ids)
	{
		return $this->db->where_in('id', $arr_ids)->get('article_material')->result_array();
	}
	/**
	\brief Метод очищает таблицу article_material от всех записей (удаляет все записи таблицы).
	*/
	function clear()
	{
		return $this->db->empty_table('article_material');
	}
}

<?php
/**
	\brief Класс модели пользователя-редактора статей.
	Данный класс описывает ввод/вывод данных таблицы user БД
*/
class User extends CI_Model
{
	public $name; ///< ФИО пользователя
  public $email; ///< Email пользователя, в данном случае используется как логин для входа в личный кабинет на сайте
  public $password; ///< Пароль пользователя, также используется для входа в личный кабинет на сайте
	public $restore_password; ///< Значение нового пароль для восстановления, заменяет значение поля password при подтверждении восстановления
  public $registration_at; ///< Дата регистрации
	public $lastlogin_at; ///< Дата последнего входа в личный кабинет
  public $is_admin = 0; ///< Логический признак администратора
	public $is_active = 1;///< Логический признак доступа пользователя к функционалу добавления и редактирования собственных статей
```

```
        public $about_me; ///< Подробная информация о авторе статьи, необязательное поле для заполнения из профиля

        public $avatar;///< Фотография пользователя, необязательное поле для заполнения из профиля

        public $work_position; ///< Должность автора статьи, необязательное поле для заполнения из профиля

        /**

        \brief Конструктор класса, который в теле вызывает конструктор родительского класса.

        */

 public function __construct()

 {

   parent::__construct();

 }

        /**

        \brief Метод добавления/вставки новых данных в таблицу

        \param[in] $arr_fields ассоциативный массив значений, в котором ключами являются наименование полей таблицы

        \return Числовой идентификатор записи в таблице

        */

        public function add($arr_fields)

 {

   foreach($arr_fields as $key => $val)

     $this->{$key} = $val;

   $r = $this->db->insert('user', $this);

   if ($r)

     return $this->db->query('SELECT LAST_INSERT_ID() as id')->result_array()[0]['id'];

   return 0;

 }

        /**

        \brief Метод обновления данных

        \param[in] $id Числовой идентификатор пользователя в таблице user

        \param[in] $arr_fields Ассоциативный массив значений, в котором ключами являются наименование полей таблицы

        \return Логический признак успешного обновления (true или false)

        */

 public function edit($id, $arr_fields)

 {

   return $this->db->where('id', $id)->update('user', $arr_fields);

 }

        /**

        \brief Метод возвращает список активных авторов

        \param[in] $num_page Номер страницы, по умолчанию 1
```

```
        \param[in] $per_page Кол-во записей авторов на страницу, по умолчанию 10

        \return Ассоциативный массив значений, в котором ключами rows и total_rows, где:

                ключ rows - это массив с ассоциативными массивами значений, где ключи являются наименование полей в таблице user

                ключи total_rows - общее кол-во активных авторов

        */

  public function get_list($num_page = 1, $per_page = 10)

  {

    $offset = ($num_page * $per_page) - $per_page;

    $ret['rows'] = $this->db->where('is_active', 1)->get('user', $per_page, $offset)->result_array();

                $ret['total_rows'] = $this->db->where('is_active', 1)->count_all('user');

    return $ret;

  }

        /**

        \brief Метод возвращает список авторов. Используется в административной части сайта

        \param[in] $num_page Номер страницы, по умолчанию 1

        \param[in] $per_page Кол-во записей авторов на страницу, по умолчанию 10

        \return Ассоциативный массив значений, в котором ключами rows и total_rows, где:

                ключ rows - это массив с ассоциативными массивами значений, где ключи являются наименование полей в таблице user

                ключи total_rows - общее кол-во авторов

        */

        public function get_list_for_admin($num_page = 1, $per_page = 10)

        {

                $offset = ($num_page * $per_page) - $per_page;

                $ret['rows'] = $this->db->get('user', $per_page, $offset)->result_array();

                $ret['total_rows'] = $this->db->count_all('user');

                return $ret;

        }

        /**

        \brief Метод возвращает данные пользователя

        \param[in] $id числовой идентификатор пользователя в таблице user

        \return Ассоциативный масссив значений с ключами полей таблицы user

        */

  public function get_by_id($id)
```

```
 {
  return $this->db->where('id', $id)->get('user')->result_array()[0];
 }
        /**
        \brief Метод возвращает данные пользователя по значению email
        \param[in] $email электронный почтовый ящик пользователя в таблице user
        \return Массив с ассоциативным массивом значений с ключами полей таблицы user
        */
        public function get_by_email($email)
        {
                return $this->db->where('email', $email)->get('user')->result_array();
        }
        /**
        \brief Метод возвращает общее кол-во пользователей
        \return Кол-во записей в таблице user
        */
        function get_count_all()
        {
                return $this->db->count_all('user');
        }
        /**
        \brief Метод активации деактивации пользователя. Используется в административной части сайта.
        \param[in] $id идентификатор пользователя в таблице user
        \param[in] $val логический признак активности - 0 или 1
        \return Логический признак успешного обновления (true или false)
        */
        function set_active_status($id, $val)
        {
                return $this->db->where('id', $id)->update('user', array('is_active' => $val));
        }
        /**
        \brief Метод записи нового пароля в поле restore_password
        \param[in] $email электронный почтовый ящик пользователя в таблице user
        \param[in] $new_password значение нового пароля прошедшее шифрование методом md5
        \return Логический признак успешного обновления (true или false)
        */
        function set_restore_password($email, $new_password)
        {
```

```
                return $this->db->where('email', $email)->update('user', array('restore_password' => $new_password));
        }
        /**
        \brief Метод изменения пароля
        \param[in] $id идентификатор пользователя в таблице user
        \param[in] $password значение пароля прошедшее шифрование методом md5
        \return Логический признак успешного обновления (true или false)
        */
        function set_password($id, $password)
        {
                return $this->db->where('id', $id)->update('user', array('password' => $password));
        }
        /**
        \brief Метод очищает таблицу user от всех записей (удаляет все записи таблицы).
        */
        function clear()
        {
                $this->db->empty_table('user');
        }
```

```
<?php
require_once(APPPATH.'/libraries/Util.php');
/**
\brief Класс контроллера пользователя (редактора статей)
*/
class CLabEditor extends CI_Controller
{
  public $data = array();///< Массив для вывода в шаблон
        /**
        \brief Метод-конструктор. Содержит инициализацию методов помошников и начальных данных
        */
  function __construct()
  {
    parent::__construct();
    $this->load->helper('url');
    $this->load->helper('cookie');
    $this->load->helper('directory');
    $this->load->helper('url');
    $this->load->helper('text');
    $this->data['current_url'] = explode('/', uri_string(current_url()));
    $this->load->library('user_agent');
    $this->data['referrer'] = $this->agent->referrer();
    if (isset($_SESSION['user_id']))
    {
                $this->data['user_id'] = $_SESSION['user_id'];
                $this->data['is_admin'] = $_SESSION['is_admin'];
                $this->data['is_active'] = $_SESSION['is_active'];
      $this->load->model('lab/User');
      $this->data['user'] = $this->User->get_by_id($this->data['user_id']);
    }
        }

        /**
        \brief Вывод страницы профиля пользователя.
        \details Выводит формы профиля и изменения пароля.
        */
        function profile()
        {
                $errors = array();
                $this->data['errors'] = array();
                if (isset($this->data['user_id']) && $this->data['user_id'] > 0)
                {
                        $this->load->model('lab/User');
```

```
                if (count($_POST) > 0 && isset($_POST['basic_info']))
                {
                    $user = $this->User->get_by_id($this->data['user_id']);
                    $arr_data = array();

        $arr_data['name'] = $this->db->escape_str($this->input->post('name'));

        $arr_data['about_me'] = $this->db->escape_str($this->input->post('about_me'));

        $arr_data['work_position'] = $this->db->escape_str($this->input->post('work_position'));

                    $path = "./uploads/avatar/{$this->data['user_id']}/";
                    $r = (!file_exists($path)) ? mkdir($path, 0777, TRUE) : TRUE;
                    $config = array(

        'upload_path' => $path,

        'allowed_types' => "gif|jpg|jpeg|png",

        'overwrite' => TRUE,

        'max_size' => "2048000",

        'max_height' => "768",

        'max_width' => "1024"

                    );
                    $this->load->library('upload', $config);
                    $this->data['upload_data'] = array();

                    $this->data['files'] = array();

                    if (count($_FILES) > 0 && isset($_FILES['avatar']) && $_FILES['avatar']['tmp_name'] != "")
                    {
                        $this->data['files'] = $_FILES;
                        if($this->upload->do_upload('avatar'))
                        {

        $this->data['upload_data']['avatar'] = $this->upload->data();

                        }
                        else {$errors['avatar'] = $this->upload->display_errors();}

                        if (count($this->data['errors']) == 0)
                        {

        // remove old picture
```

```
        $fn = $path . $user['avatar'];

        if (file_exists($fn) && !is_dir($fn))

        unlink($fn);

        $arr_data['avatar'] = $this->data['upload_data']['avatar']['orig_name'];
                                        }
                                }
                                $this->User->edit($this->data['user_id'], $arr_data);
                        }


                        if (count($_POST) > 0 && isset($_POST['change_password']))
                        {
                                $user = $this->User->get_by_id($this->data['user_id']);
                                $curr_pass = trim($this->input->post('current_password'));
                                if (strlen($curr_pass) > 0)
                                {
                                        if ($user['password'] == md5($curr_pass))
                                        {

        if (trim($this->input->post('new_password')) != '' && trim($this->input->post('new_password')) == trim($this->input->post('new_password2')))

        {

        $md5_password = md5(trim($this->input->post('new_password')));

        $this->User->set_password($this->data['user_id'], $md5_password);

        }

        else{

        $errors['new_password'] = 'Новый пароль не совпадает с подтверждённым!';

        }
                                        }
                                        else{

        $errors['curr_password'] = 'Текущий пароль не совпадает!';
                                        }
                                }
                                else{

        $errors['curr_password-1'] = 'Текущий пароль не может быть пустым!';

                                }
                        }
```

```
                $user = $this->User->get_by_id($this->data['user_id']);
                $this->data['user'] = $user;
                $this->data['errors'] = $errors;
                $this->smarty->view('author/profile.tpl', $this->data);
            }
            else
            {
                redirect( base_url());
            }
        }

        /**
        \brief Метод вывода списка статей пользователя
        \param[in] $num_page номер страницы
        */
        function articles($num_page = 1)
        {
            $this->load->model('lab/Article');
            $pag_conf['per_page'] = 10;
            $pag_conf['base_url'] = '/editor/articles';
            $pag_conf['cur_page'] = $num_page;
            $res_obj = $this->Article->get_list_by_author($this->data['user_id'], $pag_conf['per_page'], $num_page);
            $this->data['items'] = $res_obj['rows'];
            foreach($this->data['items'] as &$row)
   {
    $s = strip_tags(stripcslashes($row['content']));
                Util::str_break($s);
    $row['content'] = $s;
            }
            $pag_conf['total_rows'] = $res_obj['total_rows'];

            $quant_pages = 1;
            if ($res_obj['total_rows'] > $pag_conf['per_page'])
                $quant_pages = ceil($res_obj['total_rows']/$pag_conf['per_page']);

            $pag_conf['total_pages'] = $quant_pages;
            $this->data['paginator'] = $pag_conf;
            $this->smarty->view('author/article_list.tpl', $this->data);
        }

        /**
```

```
        \brief Вывод страницы автора, для детального просмотра

        \param[in] $user_id числовой идентификатор автора

        */

        function detail($user_id)

        {

                $this->load->model('lab/User');

                $user = $this->User->get_by_id($user_id);

                $this->data['user'] = $user;

                $this->smarty->view('author/detail.tpl', $this->data);

        }

}
```

```
<?php

require_once( APPPATH.'/libraries/Util.php' );

/**

\brief Класс контроллер общего назначения.

Данный класс реализует методы, которые доступны из основного меню для авторизованных и не авторизованных пользователей

*/

class CLab extends CI_Controller

{

  public $data = array(); ///< Массив для вывода в шаблон

        /**

        \brief Метод-конструктор. Содержит инициализацию методов помошников и начальных данных

        */

  function __construct()

  {

    parent::__construct();

    $this->load->helper('url');

    $this->load->helper('cookie');

    $this->data['current_url'] = explode('/', uri_string(current_url()));

                $this->load->model('lab/User');


//================================================================

    $dt_key = date('Y-m-d');

    $dt_access = '2017-06-11';

    if (count($_GET) > 0 && isset($_GET['dev']))

    {

      if ($_GET['dev'] == $dt_access)

      {

        set_cookie('dev', $dt_key, 0, 'alexdiz6.bget.ru');

        redirect(base_url());

      }
```

```
    }
    if (get_cookie('dev') != $dt_access)
     redirect('404');

//==============================
==============================
=====

    if (isset($_SESSION['user_id']))
    {
     $this->data['user_id'] =
$_SESSION['user_id'];
     $this->data['is_admin'] =
$_SESSION['is_admin'];
                    $this->data['is_active']
= $_SESSION['is_active'];
    }
  }
        /**
        \brief Метод вывода списка статей
        \param[in] $num_page номер
страницы
        */
        function articles($num_page = 1)
        {
                $this->load-
>model('lab/Article');
                $pag_conf['per_page'] = 10;
                $pag_conf['base_url'] =
'/articles';
                $pag_conf['cur_page'] =
$num_page;

                $this->data['element_name']
= 'article';

                $res_obj = $this->Article-
>get_list($pag_conf['per_page'],
$num_page);
                $this->data['items'] =
$res_obj['rows'];

                foreach($this->data['items']
as &$row)
    {
     $s =
nl2br(strip_tags(stripcslashes($row['descript
ion'])));
                    //Util::str_break($s);
     $row['description'] = $s;
                }

                $pag_conf['total_rows'] =
$res_obj['total_rows'];
                $quant_pages = 1;
                if ($res_obj['total_rows'] >
$pag_conf['per_page'])
                    $quant_pages =
ceil($res_obj['total_rows']/$pag_conf['per_p
age']);
```

```
		$pag_conf['total_pages'] =
$quant_pages;

		$this->data['paginator'] =
$pag_conf; //$this->pagination-
>create_links();


		$this->smarty-
>view('article/list.tpl', $this->data);

	}

	/**

	\brief Метод показа страницы
"информация"

	*/

	function info()

	{

		$this->smarty-
>view('info.tpl', $this->data);

	}

	/**

	\brief Метод вывода списка
авторов

	\param[in] $num_page номер
страницы

	*/

	function authors($num_page = 1)

	{

		$pag_conf['per_page'] = 10;

		$pag_conf['base_url'] =
'/authors';

		$pag_conf['cur_page'] =
$num_page;


		$res_obj = $this->User-
>get_list($num_page,
$pag_conf['per_page']);

		$this->data['items'] =
$res_obj['rows'];

		$pag_conf['total_rows'] =
$res_obj['total_rows'];


		$quant_pages = 1;

		if ($res_obj['total_rows'] >
$pag_conf['per_page'])

			$quant_pages =
ceil($res_obj['total_rows']/$pag_conf['per_p
age']);


		$pag_conf['total_pages'] =
$quant_pages;

		$this->data['paginator'] =
$pag_conf; //$this->pagination-
>create_links();


		$this->smarty-
>view('author/list.tpl', $this->data);

	}

	/**

	\brief Метод показа страницы
"контакты"

	*/

	function contacts()

	{
```

```
                $this->smarty->view('contacts.tpl', $this->data);
        }
        /**
        \brief Метод показа главной страницы
        */
        function index()
        {
    //Util::clear_user();
   // $this->load->model('kid/user');
   // $this->load->model('kid/task');

    $arr_test = array();

    //unset($_SESSION);
    /*
    if (isset($_SESSION) && count($_SESSION) > 0){
    foreach($_SESSION as $k => $v)
        $arr_test[$k] = $v;
    }
    */

        foreach($_REQUEST as $k => $v)
      $arr_test[$k] = $v;

    $this->data['test'] = count($arr_test) > 0 ? $arr_test: array(); //count($_REQUEST) > 0 ? $_REQUEST : array() ; //isset($_SESSION['test']) ? $_SESSION['test'] : 'undefined2';

                $this->smarty->view('index.tpl', $this->data);
        }
        /**
        \brief Метод установки данных авторизации полльзователя
        \param[in] $user ассоциативный массив данных пользователя, где ключами являются поля из таблицы user
        */
        function set_auth($user)
        {
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['is_admin'] = $user['is_admin'];
                $_SESSION['is_active'] = $user['is_active'];
                $this->User->edit($user['id'], array('lastlogin_at' => date('Y-m-d H:i:S')));
                redirect(base_url() . 'editor/profile');
        }
        /**
        \brief Вывод страницы авторизации.
        */
        function signin()
        {
```

```
        if (isset($this->data['user_id']))
            redirect(base_url() . 'editor/profile');

        $errors = array();

        if (count($_POST) > 0)
        {
            $this->data['user'] = array(
                'email' => $this->input->post('email'),
                'password' => $this->input->post('password'),
            );

            $email = $this->input->post('email');
            $res = $this->User->get_by_email($email);
            if (count($res) > 0)
            {
                $user = $res[0];
                $password = md5($this->input->post('password'));
                if ($user['password'] == $password && $user['email'] == $email)
                {
                    $this->set_auth($user);
                }
                else
                {
                    $errors[] = 'Логин и пароль не совпадают!';
                }
            }
        }
        $this->data['errors'] = $errors;
        $this->smarty->view('signin.tpl', $this->data);
    }
    /**
    \brief Вывод страницы регистрации
    */
    function signup()
    {
        $this->data['errors'] = array();

        if (isset($this->data['user_id']))
            redirect(base_url() . 'editor/profile');

        $errors = array();
        if (count($_POST) > 0)
```

```
{
if (!$this->captcha_check($this->input->post('captcha'), $this->input->post('captcha_name')))
$this->data['errors'][] = 'Неверный код капчи или срок её действия истёк!';

if (count($this->data['errors']) == 0)
{
$user = $this->User->get_by_email($this->input->post('email'));
$this->data['user'] = array(
'name' => $this->input->post('name'),
'email' => $this->input->post('email'),

'password' => $this->input->post('password'),

'password_repeat' => $this->input->post('password_repeat'),
);

if ($user){

$errors[] = 'Такой пользователь уже зарегистрирован!';
}
else
{
$pass1 = $this->input->post('password');
$pass2 = $this->input->post('password_repeat');

if($pass1 == $pass2)
{

$password = md5($pass1);

$quantity = $this->User->get_count_all();

$is_admin = ($quantity == 0) ? 1 : 0;

$user_id = $this->User->add(array(

'name' => $this->db->escape_str($this->input->post('name')),

'email' => $this->db->escape_str($this->input->post('email')),

'password' => $password,

'registration_at' => date('Y-m-d H:i:S'),

'lastlogin_at' => date('Y-m-d H:i:S'),

'is_admin' => $is_admin,
```

```
        'is_active' => ($is_admin == 1) ? 1 :
0,

        ));

        if ($user_id)

        {

        $_SESSION['user_id'] = $user_id;

        $_SESSION['is_admin'] =
$is_admin;

        $_SESSION['is_active'] =
($is_admin == 1) ? 1 : 0;

        redirect(base_url() . 'editor/profile');

        }
                                        }
                                        else
                                        {

        $errors[] = 'Пароли не совпадают.';
                                        }
                                }
                        }
                }
```

```
                $this->data['captcha'] = $this-
>captcha_init();
                $this->data['errors'] = $errors;
                $this->smarty-
>view('signup.tpl', $this->data);
        }
        /**
        \brief Вывод страницы
восстановления пароля
        */
        function restore_password()
        {
                $this->data['errors'] = array();
                if (count($_POST) > 0)
                {
                        if (!$this-
>captcha_check($this->input-
>post('captcha'), $this->input-
>post('captcha_name')))
                                $this-
>data['errors'][] = 'Неверный код капчи
или срок её действия истёк!';
                        $email = $this->input-
>post('email');
                        if (count($this-
>data['errors']) == 0)
                        {
                                if
(strlen(trim($email)) > 0)
                                {
                                        $res=
$this->User->get_by_email($email);
```

```
if ($res)
{

$pass_human = Util::rand_passwd(6);

$pass = md5($pass_human);

$this->load->library('email');

$config['protocol'] = 'sendmail';

$config['mailpath'] = '/usr/sbin/sendmail';

$config['charset'] = 'utf-8';

$config['wordwrap'] = TRUE;

$config['mailtype'] = 'html';

$this->email->initialize($config);

// $this->email->from('shket9595@list.ru', 'Разработчик');

$this->email->to($email);

$url_confirm = base_url() . "confirm_restore_password/".$res[0]['id'] . "/". $pass;

$site = substr(base_url(), 0, strlen(base_url()) -1);

$this->email->subject("Восстановление пароля на $site");

$this->email->message("Новый пароль : $pass_human<br> Подтвердите изменение пароля переходом по ссылке: <a href=\"$url_confirm\">$url_confirm</a>");

$this->email->set_alt_message("Новый пароль : $pass_human \n Подтвердите изменение пароля переходом по ссылке: $url_confirm");

// $this->email->set_header('');

if ($this->email->send())

{

$r = $this->User->set_restore_password($email, $pass);

redirect(base_url(). 'successful_restore_password');

}

}

else

{

$this->data['errors'][] = 'Нет такого пользователя!';
```

```
                }
            }
            else
            {
                $this->data['errors'][] = 'Не указан email!';
            }
        }
    }
    $this->data['captcha'] = $this->captcha_init();
    $this->smarty->view('restore_password.tpl', $this->data);
}
/**
\brief Метод подтверждения восстановления пароля.
Метод используется при подтверждении изменения пароля по ссылке из письма эл. почты.
\param[in] $uid
\param[in] $md5_restore_password
*/
function confirm_restore_password($uid, $md5_restore_password)
{
    $user = $this->User->get_by_id($uid);
    if ($user)
    {
        if ($user['restore_password'] == $md5_restore_password)
        {
            $this->User->set_password($uid, $md5_restore_password);
            $this->set_auth($user);
        }
    }
    $this->smarty->view('restore_password.tpl', $this->data);
}
/**
\brief Показ страницы об успешной отправке письма о восстановлении пароля
*/
function successful_restore_password()
{
    $this->smarty->view('author/successful_restore_password.tpl', $this->data);
}
/**
\brief Метод вывхода из личного кабинета пользователя, будучи авторизированным на сайте.
*/
function quit()
```

```
        {
   if (isset($_SESSION) &&
count($_SESSION) > 0)
     foreach ($_SESSION as $k => $v){
       unset($_SESSION[$k]);
   }
   // $this->load->helper('url');
   // $req = array();
   // if (isset($_REQUEST)){
   //   foreach($_REQUEST as $k => $v)
   //     $req[] = $k . '='. $v;
   // }
   $url = base_url();
   // if (count($req))
   //   $url .='?'. implode('&', $req);
   redirect($url);
        }
        /**
        \brief Метод получения
параметров каптчи
        \return ассоциативный массив с
ключами image и name
        где значения ключей
                image - изображение каптчи
представленое ввиде строки
закодированной методом base64
                name - имя каптчи,
округлённое число unix-время создания
каптчи
        */
        function captcha_init()
        {
                $this->load-
>library('encrypt');
                $this->load-
>helper('my_captcha');
                $vals = array(
                        'img_path' =>
'./uploads/tmp/',
                        'img_url' =>
base_url(). 'uploads/tmp/',
                        'font_path'     =>
'./static/fonts/aalbionic_bold.ttf',
                        'word_length'   => 4,
     'font_size'     => 18,
                        'img_width'     =>
'120',
     'img_height'    => 45,
     'expiration'    => 500,
                );
                $data =
create_captcha($vals);
                set_cookie('ct_'.$data['name'],
$data['time'], $vals['expiration']);

        set_cookie('cw_'.$data['name'], $this-
>encrypt->encode($data['word']),
$vals['expiration']);
                return array( 'image' =>
$data['image'], 'name' => $data['name']);
        }
        /**
```

```
        \brief Метод проверки значения
каптчи

        \param[in] $val_captcha значение
кода изображённого на каптче

        \param[in] $captcha_name имя
каптчи

        \return Логический признак
правильности введёного с каптчи кода

        */

        function
captcha_check($val_captcha,
$captcha_name)

        {

                $expiration = time();

                $this->load-
>library('encrypt');

                $cap_time =
get_cookie('ct_'.$captcha_name);

                if ($cap_time < $expiration)

                {

                        if ($this->encrypt-
>decode(get_cookie('cw_'.$captcha_name))
== $val_captcha)

                        {


        delete_cookie('cw_'.$captcha_name);


        delete_cookie('ct_'.$captcha_name);

                                return true;

                        }

                }

                delete_cookie('cw_'.$captcha_name);


                delete_cookie('ct_'.$captcha_name);

                        return false;

                }

}


<?php

require_once( APPPATH.'/libraries/Util.php'
);

/**

\brief Класс контроллера администратора

*/

class CLabAdmin extends CI_Controller

{

  public $data = array();///< Массив для
вывода в шаблон

        /**

        \brief Метод-конструктор.
Содержит инициализацию методов
помошников и начальных данных

        */

  function __construct()

  {

    parent::__construct();

    $this->load->helper('url');

    $this->load->helper('cookie');

    $this->data['current_url'] = explode('/',
uri_string(current_url()));
```

```
   $this->load->library('user_agent');
   $this->data['referrer'] = $this->agent->referrer();
   if (isset($_SESSION['user_id']))
   {
    $this->data['user_id'] = $_SESSION['user_id'];
    $this->data['is_admin'] = $_SESSION['is_admin'];
                        $this->data['is_active'] = $_SESSION['is_active'];
   }
        }
        /**
        \brief Метод вывода списка всех статей
        \param[in] $num_page номер страницы
        */
        function articles($num_page = 1)
        {
                $this->load->model('lab/Article');
                $pag_conf['per_page'] = 15;
                $pag_conf['base_url'] = '/admin/articles';
                $pag_conf['cur_page'] = $num_page;
                $res_obj = $this->Article->get_list_for_admin($pag_conf['per_page'], $num_page);
                $this->data['items'] = $res_obj['rows'];
                foreach($this->data['items'] as &$row)
   {
    $s = strip_tags(stripcslashes($row['content']));
                        Util::str_break($s);
    $row['content'] = $s;
                }
                $pag_conf['total_rows'] = $res_obj['total_rows'];

                $quant_pages = 1;
                if ($res_obj['total_rows'] > $pag_conf['per_page'])
                        $quant_pages = ceil($res_obj['total_rows']/$pag_conf['per_page']);

                $pag_conf['total_pages'] = $quant_pages;
                $this->data['paginator'] = $pag_conf;
                $this->smarty->view('admin/article_list.tpl', $this->data);
        }
        /**
        \brief Метод вывода списка всех зарегистрированных пользователей
        \param[in] $num_page номер страницы
```

```
        */
        function editors($num_page = 1)
        {
                $this->load->model('lab/User');
                $pag_conf['per_page'] = 10;
                $pag_conf['base_url'] = '/admin/editors';
                $pag_conf['cur_page'] = $num_page;
                $res_obj = $this->User->get_list_for_admin($num_page, $pag_conf['per_page']);
                $this->data['items'] = $res_obj['rows'];
                $pag_conf['total_rows'] = $res_obj['total_rows'];
                $quant_pages = 1;
                if ($res_obj['total_rows'] > $pag_conf['per_page'])
                        $quant_pages = ceil($res_obj['total_rows']/$pag_conf['per_page']);
                $pag_conf['total_pages'] = $quant_pages;
                $this->data['paginator'] = $pag_conf;
                $this->smarty->view('admin/editor_list.tpl', $this->data);
        }
        /**
        \brief Изменение признака активности автора
        \param[in] $user_id числовой идентификатор автора
        \param[in] $val числовое логическое значение активности (0 или 1)
        */
        function active_status($user_id, $val)
        {
                if ($this->data['is_admin'] == 1)
                {
                        $v = intVal($val);
                        $this->load->model('lab/User');
                        $this->User->set_active_status($user_id, $v);
                }
                redirect($this->data['referrer']);
        }
}
```

```
<?php
require_once( APPPATH.'/libraries/Util.php' );
/**
\brief Класс контроллер общего назначения.
```

```
Данный класс реализует методы, которые доступны из основного меню для авторизованных и не авторизованных пользователей

*/

class CLab extends CI_Controller

{

  public $data = array(); ///< Массив для вывода в шаблон

        /**

        \brief Метод-конструктор. Содержит инициализацию методов помошников и начальных данных

        */

  function __construct()

  {

    parent::__construct();

    $this->load->helper('url');

    $this->load->helper('cookie');

    $this->data['current_url'] = explode('/', uri_string(current_url()));

                $this->load->model('lab/User');


//==============================================================

    $dt_key = date('Y-m-d');

    $dt_access = '2017-06-11';

    if (count($_GET) > 0 && isset($_GET['dev']))

    {

      if ($_GET['dev'] == $dt_access)

      {

        set_cookie('dev', $dt_key, 0, 'alexdiz6.bget.ru');

        redirect(base_url());

      }

    }

    if (get_cookie('dev') != $dt_access)

      redirect('404');


//==============================================================


    if (isset($_SESSION['user_id']))

    {

      $this->data['user_id'] = $_SESSION['user_id'];

      $this->data['is_admin'] = $_SESSION['is_admin'];

                        $this->data['is_active'] = $_SESSION['is_active'];

    }

  }

        /**

        \brief Метод вывода списка статей

        \param[in] $num_page номер страницы

        */

        function articles($num_page = 1)
```

```
        {
                $this->load->model('lab/Article');
                $pag_conf['per_page'] = 10;
                $pag_conf['base_url'] = '/articles';
                $pag_conf['cur_page'] = $num_page;

                $this->data['element_name'] = 'article';

                $res_obj = $this->Article->get_list($pag_conf['per_page'], $num_page);
                $this->data['items'] = $res_obj['rows'];

                foreach($this->data['items'] as &$row)
   {
    $s = nl2br(strip_tags(stripcslashes($row['description'])));
                        //Util::str_break($s);
    $row['description'] = $s;
                }

                $pag_conf['total_rows'] = $res_obj['total_rows'];
                $quant_pages = 1;
                if ($res_obj['total_rows'] > $pag_conf['per_page'])
                        $quant_pages = ceil($res_obj['total_rows']/$pag_conf['per_page']);

                $pag_conf['total_pages'] = $quant_pages;
                $this->data['paginator'] = $pag_conf; //$this->pagination->create_links();

                $this->smarty->view('article/list.tpl', $this->data);
        }
        /**
        \brief Метод показа страницы "информация"
        */
        function info()
        {
                $this->smarty->view('info.tpl', $this->data);
        }
        /**
        \brief Метод вывода списка авторов
        \param[in] $num_page номер страницы
        */
        function authors($num_page = 1)
```

```
        {
                $pag_conf['per_page'] = 10;
                $pag_conf['base_url'] =
'/authors';
                $pag_conf['cur_page'] =
$num_page;

                $res_obj = $this->User-
>get_list($num_page,
$pag_conf['per_page']);
                $this->data['items'] =
$res_obj['rows'];
                $pag_conf['total_rows'] =
$res_obj['total_rows'];

                $quant_pages = 1;
                if ($res_obj['total_rows'] >
$pag_conf['per_page'])
                        $quant_pages =
ceil($res_obj['total_rows']/$pag_conf['per_p
age']);

                $pag_conf['total_pages'] =
$quant_pages;
                $this->data['paginator'] =
$pag_conf; //$this->pagination-
>create_links();

                $this->smarty-
>view('author/list.tpl', $this->data);
        }
        /**
                \brief Метод показа страницы
"контакты"
        */
        function contacts()
        {
                $this->smarty-
>view('contacts.tpl', $this->data);
        }
        /**
        \brief Метод показа главной
страницы
        */
        function index()
        {
   //Util::clear_user();
   // $this->load->model('kid/user');
   // $this->load->model('kid/task');

   $arr_test = array();

   //unset($_SESSION);
   /*
   if (isset($_SESSION) &&
count($_SESSION) > 0){
   foreach($_SESSION as $k => $v)
      $arr_test[$k] = $v;
   }
   */
```

```
        foreach($_REQUEST as $k => $v)
    $arr_test[$k] = $v;


   $this->data['test'] = count($arr_test) > 0 ? $arr_test: array(); //count($_REQUEST) > 0 ? $_REQUEST : array() ; //isset($_SESSION['test']) ? $_SESSION['test'] : 'undefined2';
                $this->smarty->view('index.tpl', $this->data);
        }
        /**
        \brief Метод установки данных авторизации полльзователя
        \param[in] $user ассоциативный массив данных пользователя, где ключами являются поля из таблицы user
        */
        function set_auth($user)
        {
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['is_admin'] = $user['is_admin'];
                $_SESSION['is_active'] = $user['is_active'];
                $this->User->edit($user['id'], array('lastlogin_at' => date('Y-m-d H:i:S')));
                redirect(base_url() . 'editor/profile');
        }
        /**
        \brief Вывод страницы авторизации.
        */
        function signin()
        {
                if (isset($this->data['user_id']))
                        redirect(base_url() . 'editor/profile');

                $errors = array();

                if (count($_POST) > 0)
                {
                        $this->data['user'] = array(
                                'email' => $this->input->post('email'),
                                'password' => $this->input->post('password'),
                                );

                        $email = $this->input->post('email');
                        $res = $this->User->get_by_email($email);
                        if (count($res) > 0)
                        {
                                $user = $res[0];
```

```
                $password =
md5($this->input->post('password'));

                if
($user['password'] == $password &&
$user['email'] == $email)

                {

                        $this-
>set_auth($user);

                }

                else

                {


        $errors[] = 'Логин и пароль не
совпадают!';

                }

            }

        }

        $this->data['errors'] = $errors;

        $this->smarty-
>view('signin.tpl', $this->data);

    }

    /**

    \brief Вывод страницы
регистрации

    */

    function signup()

    {

        $this->data['errors'] = array();


        if (isset($this-
>data['user_id']))

                redirect(base_url() .
'editor/profile');


        $errors = array();

        if (count($_POST) > 0)

        {

            if (!$this-
>captcha_check($this->input-
>post('captcha'), $this->input-
>post('captcha_name')))

                $this-
>data['errors'][] = 'Неверный код капчи
или срок её действия истёк!';


            if (count($this-
>data['errors']) == 0)

            {

                $user = $this-
>User->get_by_email($this->input-
>post('email'));

                $this-
>data['user'] = array(

                    'name'
=> $this->input->post('name'),

                    'email'
=> $this->input->post('email'),


        'password' => $this->input-
>post('password'),


        'password_repeat' => $this->input-
>post('password_repeat'),

                );
```

```
                    if ($user){

        $errors[] = 'Такой пользователь уже зарегистрирован!';

                    }

                    else

                    {

                        $pass1 = $this->input->post('password');

                        $pass2 = $this->input->post('password_repeat');

        if($pass1 == $pass2)

                        {

        $password = md5($pass1);



        $quantity = $this->User->get_count_all();

        $is_admin = ($quantity == 0) ? 1 : 0;

        $user_id = $this->User->add(array(

        'name' => $this->db->escape_str($this->input->post('name')),

        'email' => $this->db->escape_str($this->input->post('email')),

        'password' => $password,

        'registration_at' => date('Y-m-d H:i:S'),

        'lastlogin_at' => date('Y-m-d H:i:S'),

        'is_admin' => $is_admin,

        'is_active' => ($is_admin == 1) ? 1 : 0,

        ));

        if ($user_id)

        {

        $_SESSION['user_id'] = $user_id;

        $_SESSION['is_admin'] = $is_admin;

        $_SESSION['is_active'] = ($is_admin == 1) ? 1 : 0;

        redirect(base_url() . 'editor/profile');

        }

                    }

                    else

                    {

        $errors[] = 'Пароли не совпадают.';

                    }
```

```
                }
            }
        }

        $this->data['captcha'] = $this->captcha_init();
        $this->data['errors'] = $errors;
        $this->smarty->view('signup.tpl', $this->data);
    }
    /**
    \brief Вывод страницы восстановления пароля
    */
    function restore_password()
    {
        $this->data['errors'] = array();
        if (count($_POST) > 0)
        {
            if (!$this->captcha_check($this->input->post('captcha'), $this->input->post('captcha_name')))
                $this->data['errors'][] = 'Неверный код капчи или срок её действия истёк!';
            $email = $this->input->post('email');
            if (count($this->data['errors']) == 0)
            {
                if (strlen(trim($email)) > 0)
                {
                    $res= $this->User->get_by_email($email);
                    if ($res)
                    {

    $pass_human = Util::rand_passwd(6);

    $pass = md5($pass_human);

    $this->load->library('email');

    $config['protocol'] = 'sendmail';

    $config['mailpath'] = '/usr/sbin/sendmail';

    $config['charset'] = 'utf-8';

    $config['wordwrap'] = TRUE;

    $config['mailtype'] = 'html';

    $this->email->initialize($config);

    // $this->email->from('shket9595@list.ru', 'Разработчик');

    $this->email->to($email);

    $url_confirm = base_url() .
```

```
"confirm_restore_password/".$res[0]['id'] . "/". $pass;

        $site = substr(base_url(), 0, strlen(base_url()) -1);

        $this->email->subject("Восстановление пароля на $site");

        $this->email->message("Новый пароль : $pass_human<br> Подтвердите изменение пароля переходом по ссылке: <a href=\"$url_confirm\">$url_confirm</a>");

        $this->email->set_alt_message("Новый пароль : $pass_human \n Подтвердите изменение пароля переходом по ссылке: $url_confirm");

        // $this->email->set_header('');

        if ($this->email->send())

        {

        $r = $this->User->set_restore_password($email, $pass);

        redirect(base_url(). 'successful_restore_password');

        }

                                        }
                                                else

                                                {

        $this->data['errors'][] = 'Нет такого пользователя!';

                                                }
                                        }
                                        else

                                        {
                                                $this->data['errors'][] = 'Не указан email!';

                                        }
                                }
                        }

                $this->data['captcha'] = $this->captcha_init();

                $this->smarty->view('restore_password.tpl', $this->data);

        }

        /**

        \brief Метод подтверждения восстановления пароля.

        Метод используется при подтверждении изменения пароля по ссылке из письма эл. почты.

        \param[in] $uid

        \param[in] $md5_restore_password

        */

        function confirm_restore_password($uid, $md5_restore_password)
```

```
        {
                $user = $this->User->get_by_id($uid);
                if ($user)
                {
                        if ($user['restore_password'] == $md5_restore_password)
                        {
                                $this->User->set_password($uid, $md5_restore_password);
                                $this->set_auth($user);
                        }
                }
                $this->smarty->view('restore_password.tpl', $this->data);
        }
        /**
        \brief Показ страницы об успешной отправке письма о восстановлении пароля
        */
        function successful_restore_password()
        {
                $this->smarty->view('author/successful_restore_password.tpl', $this->data);
        }
        /**
        \brief Метод вывхода из личного кабинета пользователя, будучи авторизированным на сайте.
        */
        function quit()
        {
   if (isset($_SESSION) && count($_SESSION) > 0)
     foreach ($_SESSION as $k => $v){
       unset($_SESSION[$k]);
   }
   // $this->load->helper('url');
   // $req = array();
   // if (isset($_REQUEST)){
   //   foreach($_REQUEST as $k => $v)
   //     $req[] = $k . '='. $v;
   // }
   $url = base_url();
   // if (count($req))
   //   $url .='?'. implode('&', $req);
   redirect($url);
        }
        /**
        \brief Метод получения параметров каптчи
        \return ассоциативный массив с ключами image и name
        где значения ключей
```

```
        image - изображение каптчи представленое ввиде строки закодированной методом base64

        name - имя каптчи, округлённое число unix-время создания каптчи

    */

    function captcha_init()

    {

        $this->load->library('encrypt');

        $this->load->helper('my_captcha');

        $vals = array(

            'img_path' => './uploads/tmp/',

            'img_url' => base_url(). 'uploads/tmp/',

            'font_path'     => './static/fonts/aalbionic_bold.ttf',

            'word_length'   => 4,

    'font_size'     => 18,

            'img_width'     => '120',

    'img_height'    => 45,

    'expiration'    => 500,

        );

        $data = create_captcha($vals);

        set_cookie('ct_'.$data['name'], $data['time'], $vals['expiration']);

        set_cookie('cw_'.$data['name'], $this->encrypt->encode($data['word']), $vals['expiration']);

            return array( 'image' => $data['image'], 'name' => $data['name']);

    }

    /**

    \brief Метод проверки значения каптчи

    \param[in] $val_captcha значение кода изображённого на каптче

    \param[in] $captcha_name имя каптчи

    \return Логический признак правильности введёного с каптчи кода

    */

    function captcha_check($val_captcha, $captcha_name)

    {

        $expiration = time();

        $this->load->library('encrypt');

        $cap_time = get_cookie('ct_'.$captcha_name);

        if ($cap_time < $expiration)

        {

            if ($this->encrypt->decode(get_cookie('cw_'.$captcha_name)) == $val_captcha)

            {
```

```
        delete_cookie('cw_'.$captcha_name);

        delete_cookie('ct_'.$captcha_name);
                                return true;
                        }
                }

        delete_cookie('cw_'.$captcha_name);

        delete_cookie('ct_'.$captcha_name);
                return false;
        }
}


<?php
/**
\brief Класс контроллера статьи
*/
class CLabArticle extends CI_Controller
{
  public $data = array();///< Массив для вывода в шаблон
        /**
        \brief Метод-конструктор. Содержит инициализацию методов помошников и начальных данных
        */
  function __construct()
  {
    parent::__construct();
    $this->load->helper('url');
    $this->load->helper('cookie');
    $this->load->helper('directory');
    $this->load->helper('url');
    $this->load->helper('text');
    $this->data['current_url'] = explode('/', uri_string(current_url()));
    $this->load->library('user_agent');
    $this->data['referrer'] = $this->agent->referrer();
                $this->data['errors'] = array();
    if (isset($_SESSION['user_id']))
    {
                        $this->data['user_id'] = $_SESSION['user_id'];
                        $this->data['is_admin'] = $_SESSION['is_admin'];
                        $this->data['is_active'] = $_SESSION['is_active'];
    }
    $this->data['BASEPATH'] = BASEPATH;
        }
        /**
        \brief Вывод формы добавления статьи
        */
        function add()
```

```
        {
                if (!isset($this->data['user_id']))
                        redirect( base_url() .'articles');

                if ($this->data['user_id'])
                {
                        $this->load->model('lab/User');
                        $user = $this->User->get_by_id($this->data['user_id']);
                        if ($user['is_active'] == 0)
                                redirect($this->data['referrer']);
                }

                $this->data['action_name'] = 'add';
                $this->data['post'] = isset($_POST) ? $_POST : array();
                $this->data['files'] = isset($_FILES) ? $_FILES : array();

                if (count($_POST) > 0)
                {
                        $content = $this->db->escape_str($this->input->post('content'));
    if (mb_strlen($content) > 0)
    {
                                $article_id = null;
                                $this->load->model('lab/Article');
                                $article_id = $this->Article->add(
                                        array(
        'title' => $this->db->escape_str($this->input->post('title')),
        'author_id' => $this->data['user_id'],
        'content' => $content,
        'create_at' => date('Y-m-d H:i:S'),
        'is_active' => isset($_POST['is_active']) ? 1 : 0 ,
        'keywords' =>  $this->db->escape_str($this->input->post('keywords')),
        'description' => $this->db->escape_str($this->input->post('description')),
        )
                                );
                                $this->upload_files($article_id);
```

```
                    if (count($this->data['errors']) == 0)
        redirect(base_url(). 'editor/articles');
                    }
                    else
                    {
                        $this->data['detail']['title'] = $this->db->escape_str($this->input->post('title'));
                    }
                }
                $this->smarty->view('article/form.tpl', $this->data);
        }
        /**
        \brief Вывод формы редактирования статьи
        \param[in] $article_id Числовой идентификатор статьи в таблице БД article
        */
        function edit($article_id)
        {
                $this->data['action_name'] = 'edit';
                $this->load->model('lab/Article');
                $this->load->model('lab/ArticleMaterial');
                $author_id = intval($this->Article->get_author($article_id));
                if (!isset($this->data['user_id']) || ($author_id != $this->data['user_id']))
                        redirect( base_url() .'articles');
                if ($this->data['user_id'])
                {
                        $this->load->model('lab/User');
                        $user = $this->User->get_by_id($this->data['user_id']);
                        if ($user['is_active'] == 0)
                                redirect($this->data['referrer']);
                }
                if (count($_POST) > 0)
                {
                        $content = $this->db->escape_str($this->input->post('content'));
                        if (mb_strlen($content) > 0)
                        {
                                $article_data = $this->Article->get_detail($article_id);
                                $ret = $this->Article->edit(
                                        $article_id,
                                        array(
                                                'title' => $this->db->escape_str($this->input->post('title')),
```

```
        'author_id' => $this->data['user_id'],

        'content' => $content,

        'modified_at' => date('Y-m-d H:i:S'),

        'create_at' =>
$article_data['create_at'],

        'is_active' =>
isset($_POST['is_active']) ? 1 : 0 ,

        'keywords' =>  $this->db-
>escape_str($this->input-
>post('keywords')),

        'description' => $this->db-
>escape_str($this->input-
>post('description')),

                                ));

                                // удаление
отмеченных файлов

        $arr_materials_remove = array();

                                if
(isset($_POST['material-id']))

                                {


        $arr_materials_remove = $this-
>input->post('material-id');

                                    if
(count($arr_materials_remove) > 0)

                                    {

        $arr_materials = $this-
>ArticleMaterial-
>get_list_by_arr($arr_materials_remove);

        foreach($arr_materials as $item)

        {

        $fn =
"./uploads/materials/$article_id/".
$item['val'].$item['file_ext'];

        if (file_exists($fn))

        {

                $this->ArticleMaterial-
>remove($item['id']);

                unlink($fn);

        }

        }

                                        }

                                }

                                $this-
>upload_files($article_id);

                                if (count($this-
>data['errors']) == 0)

        redirect(base_url(). 'editor/articles');

                        }
```

```
                $this->data['referrer'] = $this->input->post('referrer');
            }
            $article_data = $this->Article->get_detail($article_id);
            $article_data['title'] = nl2br(strip_tags(stripcslashes($article_data['title'])));
            $article_data['content'] = stripcslashes($article_data['content']);
            $article_data['description'] = strip_tags(stripcslashes($article_data['description']));
            if (isset($article_data['materials']))

        $article_data['materials'] = json_decode('[' . $article_data['materials'] . ']');
            $this->data['post'] = isset($_POST) ? $_POST : array();
            $this->data['files'] = isset($_FILES) ? $_FILES : array();
            $this->data['detail'] = $article_data;
            $this->smarty->view('article/form.tpl', $this->data);
        }
        /**
        \brief Метод удаления статьи
        \param[in] $article_id Числовой идентификатор статьи в таблице БД article
        */
        function remove($article_id)
        {
            $this->load->model('lab/Article');
            $this->load->model('lab/ArticleMaterial');
            $author_id = $this->Article->get_author($article_id);
            $result_template = "article/error_removed.tpl";
            $this->data['removed_article'] = $article_id;
            if (isset($this->data['user_id']) && ($author_id == $this->data['user_id'] || $this->data['is_admin'] == 1))
            {
                $r = $this->Article->remove($article_id);
                if ($r)
                {
                    $this->ArticleMaterial->remove_by_article($article_id);

        $result_template = "article/succesful_removed.tpl";
                }
            }
            $this->smarty->view($result_template, $this->data);
        }
```

```
        /**
        \brief Вывод статья для детального просмотра
        \param[in] $article_id Числовой идентификатор статьи в таблице БД article
        */
        function detail($article_id)
        {
                $this->load->model('lab/Article');
                $article_data = $this->Article->get_detail($article_id);
                $article_data['title'] = strip_tags(stripcslashes($article_data['title']));
                $article_data['content'] = stripcslashes($article_data['content']);
                $article_data['keywords'] = strip_tags(stripcslashes($article_data['keywords']));
                $article_data['description'] = strip_tags(stripcslashes($article_data['description']));
                $arr_images = array();
                if (isset($article_data['materials'])){

        $article_data['materials'] = json_decode('[' . $article_data['materials'] . ']');
    foreach($article_data['materials'] as $img)
      if ($img->i == 1)
          $arr_images[] = $img->val;
    }
                $this->data['detail'] = $article_data;
   $this->data['detail']['images'] = $arr_images;
                $this->smarty->view('article/detail.tpl', $this->data);
        }
        /**
        \brief Метод загрузки дополнительных материалов статьи
        \param[in] $article_id Числовой идентификатор статьи в таблице БД article
        */
        function upload_files($article_id)
        {
                // загрузка новых файлов, если есть.
                if ($article_id && count($_FILES) > 0)
                {
                        $this->load->model('lab/ArticleMaterial');
                        $path = "./uploads/materials/$article_id/";
                        $r = (!file_exists($path)) ? mkdir($path, 0777, TRUE) : true;
                        $config = array(
```

```
                    'upload_path' => $path,

        //'allowed_types' => "gif|jpg|jpeg|png|iso|dmg|zip|rar|doc|docx|xls|xlsx|ppt|pptx|csv|ods|odt|odp|pdf|rtf|sxc|sxi|txt|exe|avi|mpeg|mp3|mp4|3gp",

                    'allowed_types' => "gif|jpg|jpeg|png|zip|rar|doc|docx|xls|xlsx|ppt|pptx|csv|ods|odt|odp|pdf|rtf|sxc|sxi|txt",

                    'overwrite' => TRUE,

                    'max_size' => "2048000", // Can be set to particular file size , here it is 2 MB(2048 Kb)

                    'max_height' => "768",

                    'max_width' => "1024"

                );

                $this->load->library('upload', $config);

                $this->data['upload_data'] = array();

                $this->data['errors'] = array();

                foreach ($_FILES as $field_name => $val)

                {

                    if (isset($val['tmp_name']) && trim($val['tmp_name']) != '')

                    {

                        if($this->upload->do_upload($field_name))

                        {

        $this->data['upload_data'][$field_name] = $this->upload->data();

                        }

                        else

                        {

        $this->data['errors'][$field_name] = $this->upload->display_errors();

                        }

                    }

                }

                foreach ($this->data['upload_data'] as $field_name => $data)

                {

                    $title = $this->db->escape_str($this->input->post($field_name. '-title'));

                    $this->ArticleMaterial->add(array(

        'article_id' => $article_id,

                    'title' => $title,

                    'val' => $data['raw_name'],

        'file_ext' => $data['file_ext'],
```

```
            'is_image' => $data['is_image'],
                                                'sort' =>
0, //(intval($sort) > 0) ? intval($sort) : 10,
                                            ));
                                    }
                            }
                    }
            }
```